NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PRELIMINARY DESIGN CODE FOR AN AXIAL STAGE COMPRESSOR

by

Rizwan R. Ramakdawala

September 2001

Thesis Advisor: Raymond P. Shreeve
Second Reader: Garth V. Hobson

Approved for public release; distribution is unlimited.

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
30 Sep 2001	N/A	-

Title and Subtitle Preliminary Design Code for an Axial Stage Compressor	**Contract Number**
	Grant Number
	Program Element Number
Author(s) Rizwan R. Ramakdawala	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Research Office Naval Postgraduate School Monterey, Ca 93943-5138	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es)	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes	
Abstract	
Subject Terms	
Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 136	

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2001	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Title (Mix case letters) Preliminary Design Code for an Axial Stage Compressor	**5. FUNDING NUMBERS**
6. AUTHOR(S) Rizwan R. Ramakdawala	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(maximum 200 words)***

Current two-dimensional preliminary design codes use structured programming, which is rigid and does not allow the user to vary parameters easily. This study uses object-oriented programming to allow the user to vary all selectable parameters in a familiar Windows operating environment. The programmed design is based on the assumptions of axial and free-vortex flow between blade rows, simple radial equilibrium, and a thermally and calorically perfect gas. The program allows a fan or core stage design and uses an open architecture to facilitate upgrades and extensions.

Using the Naval Postgraduate School's (NPS) transonic compressor design as input, the preliminary design code output was compared to the detailed throughflow design of the transonic compressor. The results agreed reasonably well with detailed throughflow design. With some minor improvements this code can easily be used to develop a preliminary design that can be optimized to the user's requirements.

14. SUBJECT TERMS Axial Compressor Design, Compressor Preliminary Design Code			**15. NUMBER OF PAGES** 138
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

PRELIMINARY DESIGN CODE FOR AN AXIAL STAGE COMPRESSOR

Rizwan R. Ramakdawala
B.S., University of Maryland, College Park, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2001

Author: Rizwan R. Ramakdawala

Approved by: Raymond P. Shreeve, Thesis Advisor

Garth V. Hobson, Second Reader

Max F. Platzer, Chairman
Department or Aeronautics and Astronautics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Current two-dimensional preliminary design codes use structured programming, which is rigid and does not allow the user to vary parameters easily. This study uses object-oriented programming to allow the user to vary all selectable parameters in a familiar Windows operating environment. The programmed design is based on the assumptions of axial and free-vortex flow between blade rows, simple radial equilibrium, and a thermally and calorically perfect gas. The program allows a fan or core stage design and uses an open architecture to facilitate upgrades and extensions.

Using the Naval Postgraduate School's (NPS) transonic compressor design as input, the preliminary design code output was compared to the detailed throughflow design of the transonic compressor. The results agreed reasonably well with detailed throughflow design. With some minor improvements this code can easily be used to develop a preliminary design that can be optimized to the user's requirements.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF SYMBOLS

Symbols

α	Absolute flow angle
β	Relative flow angle
δ	Tip gap
δ^*	Deviation angle
Φ	Flow parameter
ϕ	Axial velocity ratio
ϕ^*	Camber angle
γ	Ratio of specific heats
η	Efficiency
ι^*	Incidence angle
$\dot{m}$	Mass flow
Π	Total pressure ratio
σ	Solidity
τ	Total temperature ratio
ω	Rotor angular velocity
$\tilde{w}$	Loss coefficient
A	Area
AR	Aspect ratio
C	Chord
D	Diffusion factor
f_σ	Solidity fraction
g*	Gravitational constant in units conversion
J	Mechanical equivalent of heat
H	Blade height
M	Mach number
P	Pressure
R	Gas constant or mean-line ratio
r	Radius
r_{st}	Degree of reaction
S	Blade spacing
T	Temperature
t/c	Thickness to chord ratio
V	Velocity in stator frame of reference
W	Velocity in rotating frame of reference
X	Velocity as a fraction of inlet limiting velocity
Y	Velocity as a fraction of the local limiting velocity
Z	Number of blades

Subscripts

θ	Tangential component
1	Inlet
2	Rotor
3	Stator
21	Ratio of rotor exit to rotor inlet
32	Ratio of stator exit to stator inlet
31	Ratio of stator exit to rotor inlet
E	Equivalent (ideal rotor outlet [Ref. 8])
H	Hub
ht	Hub-to-tip ratio
m or mn	Mean
p	Profile
R	Relative
rev	Revised
s	Shock
sftc	Secondary flow and tip clearance
T	Total
t	Tip or stagnation
U	Wheel speed component
W	Relative component
Z	Axial component

I. INTRODUCTION

The design of a new axial compressor involves a sequence of steps, progressing through a sequence of computational programs of increasing complexity and sophistication. The first, or 'preliminary design' step, can be a one-dimensional 'mean-line calculation', or a two-dimensional calculation of a preliminary flow path and selection of the blading. The latter is the minimum required if the overall task is the preliminary design of an aircraft gas turbine engine. It is also what is required to provide the input for detailed throughflow design codes [Ref. 1], which, in turn, generate inputs to codes, which compose the blade geometry [Ref. 2] for manufacturing.

The current preliminary compressor design code used in aircraft engine design courses [Ref. 3], was developed progressively using different versions of Hewlett-Packard (HP) BASIC [Ref. 4]. This highly structured programming language is rigid and does not allow the user to vary one, or several, parameters easily in order to change or optimize a design. Use of the programs requires the installation of HP BASIC for Windows.

The purpose of the present study was to develop a preliminary compressor design code that would satisfy the following conditions:

- Be simple to use.
- Allow all selectable parameters to be changed.
- Require only the Windows operating system.
- Use an open architecture to allow upgrades (e.g., different loss models) or additions (e.g., turbine design)

To meet these conditions, Microsoft's Visual Basic 6.0 [Ref. 5] was selected and used to develop the source code.

THIS PAGE INTENTIONALLY LEFT BLANK

II. PROGRAM OVERVIEW

A. ASSUMPTIONS

The following assumptions were made in the development of the programmed equations and design of the code:

- Design is for an *axial flow* compressor (with radial movement of the mean line).
- *Simple* radial equilibrium is assumed from hub to tip.
- Free vortex flow is assumed between blade rows.
- A conceptual engine design study [using Ref. 6 for example] will generate the inputs for the code.

B. INPUTS AND OUTPUTS

The stage is shown schematically in Fig. 1, and the inputs required by the code, are shown in Table 1.

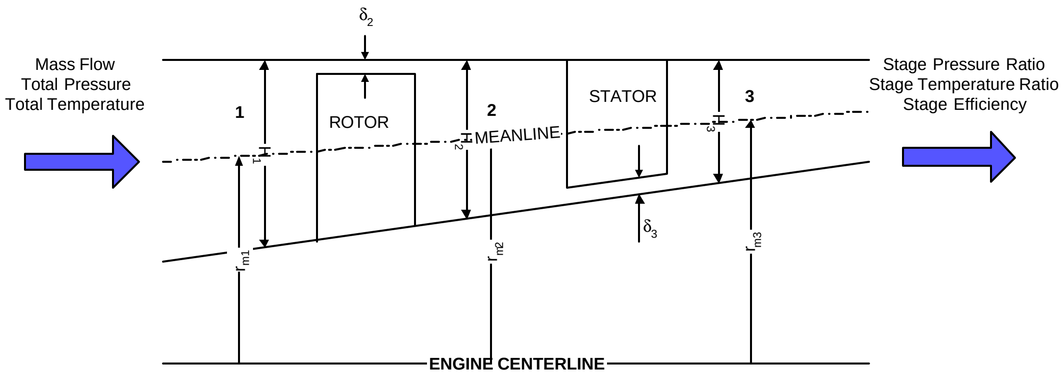

Figure 1. Throughflow Schematic

Inlet (1)	Rotor (2)	Stator (3)
$\dot{m}$	D_{2m}	A_{31}
P_t	σ_{2m}	σ_{3m}
T_t	R_{21}	R_{32}
R	ϕ_{21}	ϕ_{32}
γ	δ_2	δ_3
ω	AR_2	AR_3
M_{Z1t}	t/c_{2h}	t/c_{3h}
M_{W1t}	t/c_{2m}	t/c_{3m}
α_{1t}	t/c_{2t}	t/c_{3t}
	$f_{\sigma 2}$	$f_{\sigma 3}$
	Z_2	Z_3

Table 1. User-Provided Inputs

At the inlet station, it is required to specify the gas (through the gas constant and ratio of specific heats; a thermally and calorically perfect gas is assumed), the mass flow rate, and the stagnation conditions. Then, in order to accommodate the usual fan or core design constraints, four parameters must be specified for the rotor; namely, rotational speed, axial Mach number, relative Mach number, and flow angle at the tip. Only three of these four are independent. Off-line calculation is required to determine the other one.

The design selections for the rotor and stator are listed in columns two and three of Table 1. Note that the blade loading throughout the stage is determined by the selection of the diffusion factor at only one location for one blade. Blade aspect ratio and thickness variations are choices, which are determined by structural considerations. Structural constraints are not included in the program.

The parameters, which are calculated and output by the code, are shown in Table 2.

	Inlet (1)			**Rotor (2)**			**Stator (3)**		
Comp.	**Hub (h)**	**Mean (m)**	**Tip (t)**	**Hub (h)**	**Mean (m)**	**Tip (t)**	**Hub (h)**	**Mean (m)**	**Tip (t)**
X	X_{1h}	X_{1m}	X_{1t}	X_{2h}	X_{2m}	X_{2t}	X_{3h}	X_{3m}	X_{3t}
$\mathbf{X_Z}$	X_{Z1h}	X_{Z1m}	X_{Z1t}	X_{Z2h}	X_{Z2m}	X_{Z2t}	X_{Z3h}	X_{Z3m}	X_{Z3t}
$\mathbf{X_U}$	X_{U1h}	X_{U1m}	X_{U1t}	X_{U2h}	X_{U2m}	X_{U2t}	X_{U3h}	X_{U3m}	X_{U3t}
$\mathbf{X_W}$	X_{W1h}	X_{W1m}	X_{W1t}	X_{W2h}	X_{W2m}	X_{W2t}	X_{W3h}	X_{W3m}	X_{W3t}
$\mathbf{X_q}$	$X_{\theta 1h}$	$X_{\theta 1m}$	$X_{\theta 1t}$	$X_{\theta 2h}$	$X_{\theta 2\mu}$	$X_{\theta 2t}$	$X_{\theta 3h}$	$X_{\theta 3m}$	$X_{\theta 3t}$
r	r_{1h}	r_{1m}	r_{1t}	r_{2h}	r_{2m}	r_{2t}	r_{3h}	r_{3m}	r_{3t}
M	M_{1h}	M_{1m}	M_{1t}	M_{2h}	M_{2m}	M_{2t}	M_{3h}	M_{3m}	M_{3t}
$\mathbf{M_Z}$	M_{Z1h}	M_{Z1m}	M_{Z1t}	M_{Z2h}	M_{Z2m}	M_{Z2t}	M_{Z3h}	M_{Z3m}	M_{Z3t}
$\mathbf{M_W}$	M_{W1h}	M_{W1m}	M_{W1t}	M_{W2h}	M_{W2m}	M_{W2t}	M_{W3h}	M_{W3m}	M_{W3t}
b	β_{1h}	β_{1m}	β_{1t}	β_{2h}	β_{2m}	β_{2t}	β_{3h}	β_{3m}	β_{3t}
a	α_{1h}	α_{1m}	Input	α_{2h}	α_{2m}	α_{2t}	α_{3h}	α_{3m}	α_{3t}
Y				Y_{2h}	Y_{2m}	Y_{2t}	Y_{3h}	Y_{3m}	Y_{3t}
$\mathbf{Y_W}$				Y_{W2h}	Y_{W2m}	Y_{W2t}	Y_{W3h}	Y_{W3m}	Y_{W3t}
D				D_{2h}	Input	D_{2t}	D_{3h}	D_{3m}	D_{3t}
s				σ_{2h}	Input	σ_{2t}	σ_{3h}	Input	σ_{3t}
$\mathbf{r_{st}}$				r_{st2h}	r_{st2m}	r_{st2t}	r_{st3h}	r_{st3m}	r_{st3t}
i*				ι^*_{2h}	ι^*_{2m}	ι^*_{2t}	ι^*_{3h}	ι^*_{3m}	ι^*_{3t}
f*				ϕ^*_{2h}	ϕ^*_{2m}	ϕ^*_{2t}	ϕ^*_{3h}	ϕ^*_{3m}	ϕ^*_{3t}
d*				δ^*_{2h}	δ^*_{2m}	δ^*_{2t}	δ^*_{3h}	δ^*_{3m}	δ^*_{3t}
$\mathbf{r_{ht}}$		r_{ht1}			r_{ht2}			r_{ht3}	
A		A_1			A_2			A_3	
$\mathbf{T/T_{t1}}$		T_1/T_{t1}			T_2/T_{t1}			T_3/T_{t1}	
$\mathbf{P/P_{t1}}$		P_1/P_{t1}			P_2/P_{t2}			P_3/P_{t3}	
$\mathbf{T_t/T_{t1}}$					T_{t2}/T_{t1} (τ)			T_{t3}/T_{t1}	
$\mathbf{P_t/P_{t1}}$					P_{t2}/P_{t1}			P_{t3}/P_{t1} (Π)	
$\mathbf{A/A_1}$					A_2/A_1			A_3/A_1	
$w_\mathbf{p}$					ω_{p2}			ω_{p3}	
$w_\mathbf{sftc}$					ω_{sftc2}			ω_{sftc3}	
$w_\mathbf{s}$					ω_{s2}			ω_{s3}	
$w_\mathbf{T}$					ω_{T2}			ω_{T3}	
F					Φ_{2m}			Φ_{3m}	
H					H_2			H_3	
C					C_2			C_3	
S					S_2			S_3	
$\mathbf{Z_{rev}}$					Z_{rev2}			Z_{rev3}	
$\mathbf{AR_{rev}}$					AR_{rev2}			AR_{rev3}	
$\mathbf{C_{rev}}$					C_{rev2}			C_{rev3}	

Table 2. Outputs

THIS PAGE INTENTIONALLY LEFT BLANK

III. ALGORITHMS

A. PROGRAM SEQUENCE

The overall program sequence is shown in Fig. 2. After the program is started a "splash" screen is first shown. A splash screen is an introductory screen (similar to an "about" screen) which states the name, owner(s), and version of the program. Next, an interactive screen appears, giving the user a choice of either going through a compressor design or a turbine design. The scope of this paper covers only a compressor design; therefore the turbine option has been disabled. When the *Compressor Design* button is pressed the input screen will appear. The design sequence is shown in Fig. 3. The required inputs are shown in Table 1. Once the inputs are typed in the user presses the *OK* button and the main screen appears. The main screen is laid out in a tab format with five tabs. The tabs are as follows:

- Inlet Conditions (1)
- Rotor Calculations (2)
- Stator Calculations (3)
- Stage Performance
- Blade Geometry

This allows the variables and code to be grouped so the user can better understand what is being displayed. It also allows the programmer to develop the code with an open architecture for easy updates. At this point the user can still modify the inputs before executing the design calculations.

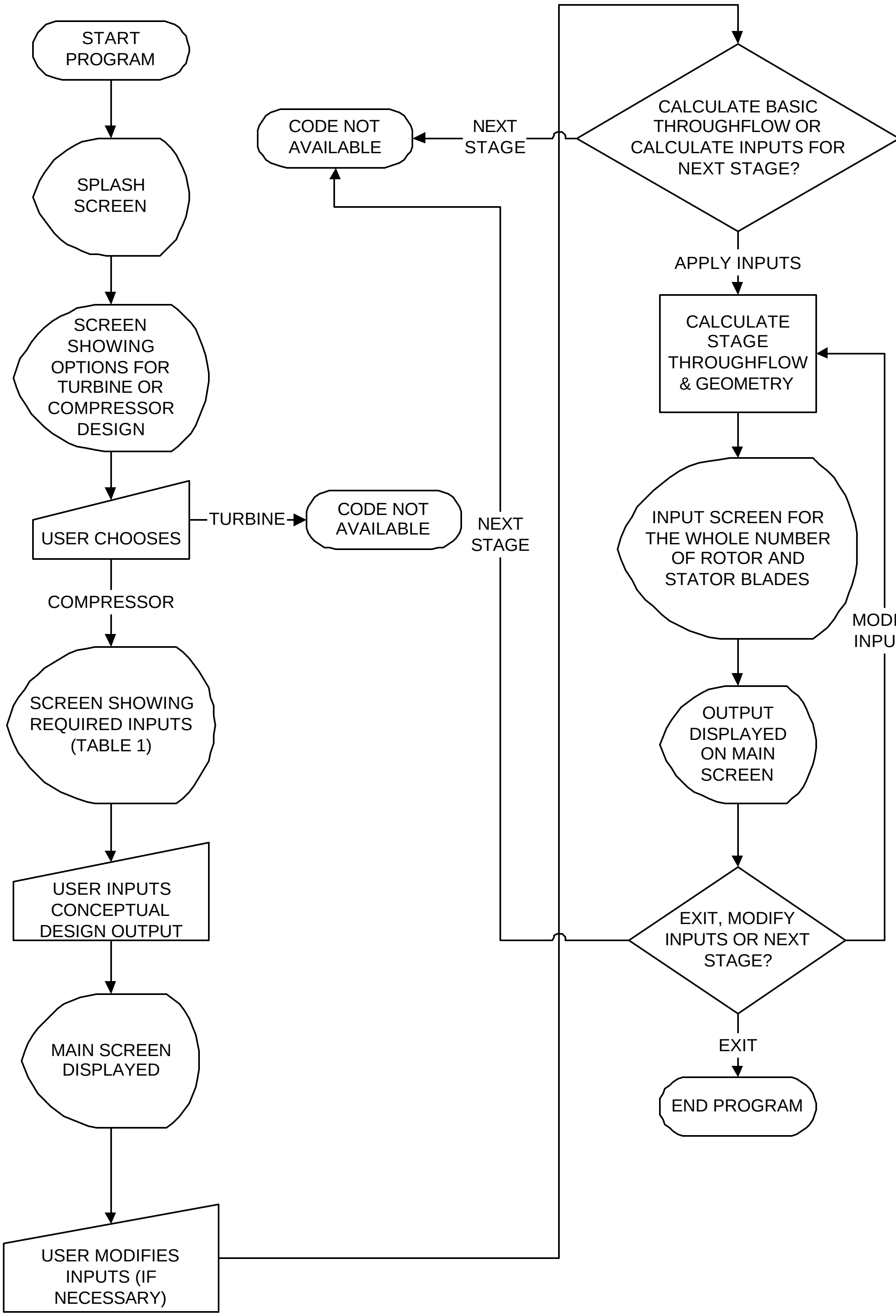

Figure 2. Main Program Flowchart

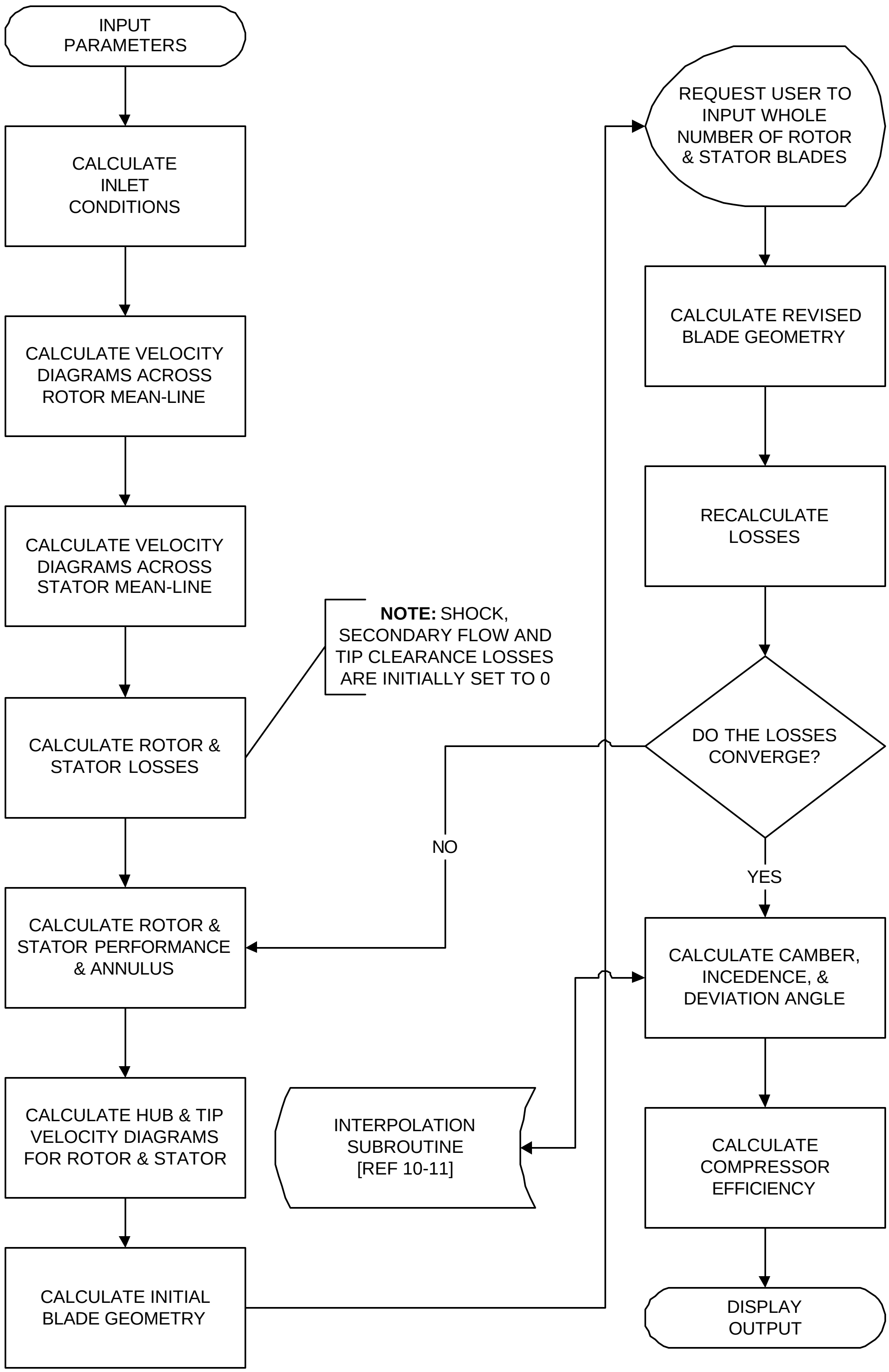

Figure 3. Design Flowchart

The user has the option to either apply the inputs to the design calculations or calculate the next stage. A multistage approach is also outside the scope of this paper so this option has been disabled. After the user presses the *Apply* button the program calculates the basic throughflow velocity diagrams, losses, performance and geometry. During the throughflow calculations an input screen is displayed for the user to input the whole number of blades needed (Z) for the rotor and stator. The main screen is displayed again with the outputs filled in the text boxes. The outputs are shown in Table 2. With the initial throughflow calculations complete the user may end the program of modify the inputs until the desired results are achieved.

B. DESIGN EQUATIONS

The equations programmed in the source code are listed in Appendix A. The equations are grouped the same as the tabs on the main screen. The equations for the velocity diagrams, profile loss, secondary flow loss, tip clearance loss, and stage performance are from Shreeve [Ref. 7 and 8]. The shock loss equation is from Koch and Smith [Ref. 9].

The incidence, deviation and camber angles (which relate the flow angles to the blade geometry, as shown in Fig. 4) were derived from NASA SP-36 [Ref. 10]. Sixth degree polynomial curvefits were used to approximate the data in Figs. 137, 138, 142, 161, 162, 168, 172, 178, 179, and 180 of Ref. 10. An interpolation routine [Ref. 11], was programmed to solve for unknowns within the curve-fits.

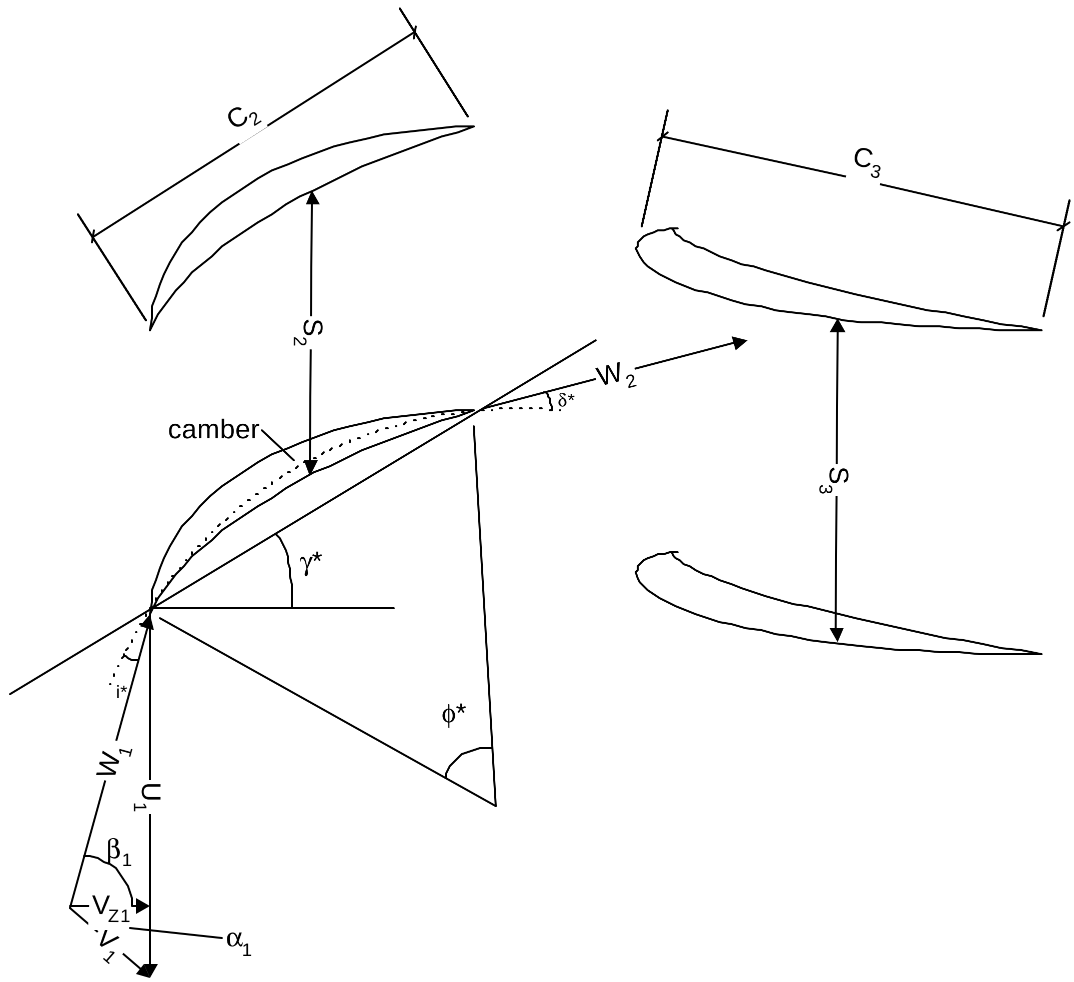

Figure 4. Blade Geometry Schematic

The performance of the stage is calculated following the thermodynamic process shown in Fig. 5. The conditions on the mean line are taken as being representative of the stage; however, the loss coefficients include contributions due to secondary flow and tip-clearance.

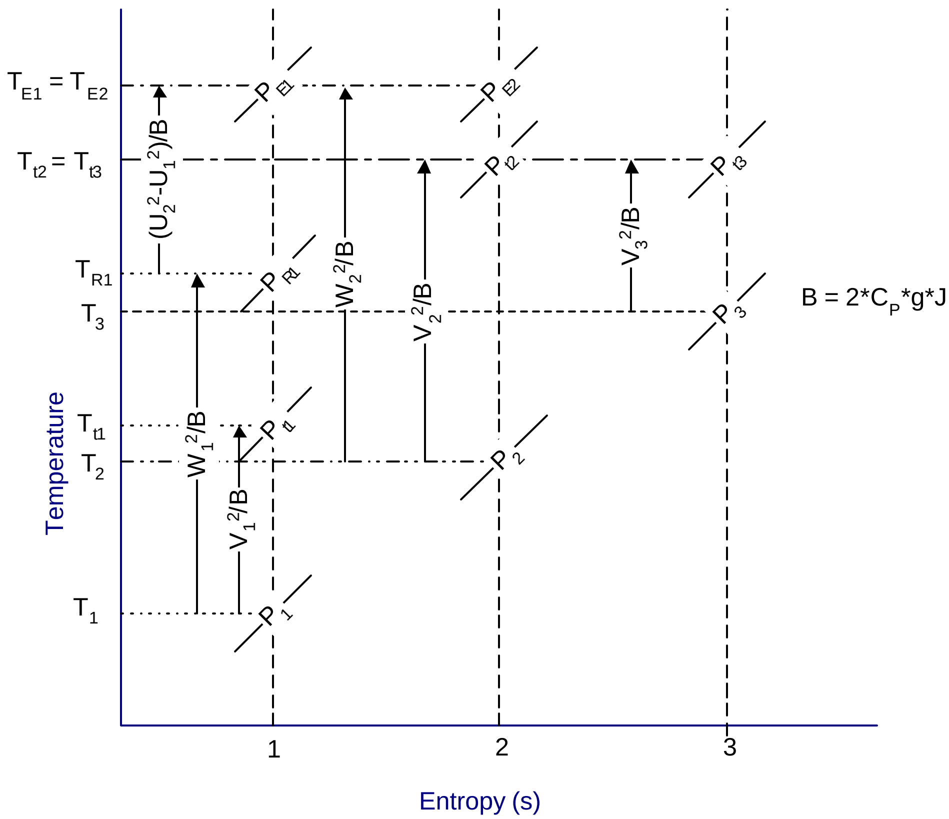

Figure 5. T-s Diagram for a Compressor Stage

IV. PROGRAM STRUCTURE

A. OBJECTS

Forms (a.k.a. screens) are used to interact with the user. Forms are 'objects' in Visual Basic and Visual C++ programming. Other common objects include text boxes, option boxes and tabs. Objects have properties (this is how the object looks to the user) which can be affected during design or run time (program execution). The objects allow the user to modify the inputs throughout the program and either executes the desired modification or keeps the original inputs. The objects (or "screen captures") are shown in Appendix B.1.

B. VARIABLES

Variables are used in programming as placeholders of data where they are used throughout the code to execute statements. Two-dimensional arrays are used extensively throughout the code since they allow a parameter that changes both throughflow and spanwise to be stored in one variable. For example, X_W (relative velocity) varies from inlet (1) to stator (3) and from hub (1) to tip (3). If we store this variable separately we would need to track nine variables. Instead, we simply track one variable $X_W(3,3)$ where the first number is the throughflow number and the second number is the spanwise number. Appendix B.2 shows all the variables used in the program. This includes all the input and output parameters as well as others that are not seen by the user.

C. MODULES

Code is used to state what needs to be executed. The code syntax is based on BASIC, which translates "pseudo" language to machine language. Code is written for objects to tell the object what to do when acted upon by the user. For this program there is very little code for objects. Most of the code is a sequence of design equations which do not cause changes in any objects and require minimal interaction with the user during execution. Modules are used in this situation. Modules take subroutines or functions that do not affect an object and keep them in a separate file for easier reading during

programming or debugging. The code, under the form ‘frmCompressor’, is the design code sequence shown in Figure 2. Appendix B.3 shows all the source code for the program.

V. RESULTS AND DISCUSSION

Sanger's transonic compressor design [Ref. 13] was used to test the ability of the code to approximate a known axial stage design. A set of hand calculations was also carried out independently of the code in order to both validate the coding, and to document the test case. The design input flow conditions, and parameters derived from the final geometry of the Sanger design, were used as inputs to both the preliminary design code and hand calculations. Appendix C.1 and Appendix C.2 document, in detail, the results of the hand calculations and the preliminary design code, respectively.

The results from hand calculations and from the code were compared to the output of the streamline curvature code applied to the Sanger design, which is given in Appendix E of Sanger [Ref. 13]. The comparisons are shown in Appendix D. It can be seen from the comparison charts in Appendix D that the hand calculations agreed fully with the preliminary design code calculations. Also, for most parameters, the hand and preliminary design code calculations agreed with the streamline curvature code outputs. In Figure D1, the calculated annulus geometry agrees well at the inlet and then begins to deviate somewhat through the stage. This is because the streamline curvature code takes into account blockage, whereas the preliminary design code and hand calculations, do not. This is easy to correct. The stage performance however, shown in Fig. D2, is predicted very successfully by the code. In Fig. D3 and D4, the differences in solidity and blade height are also, indirectly, the result of omitting blockage from the calculation of annulus area. Velocity diagram details are compared in Figs. D5 to D8. It is clear that the preliminary design code reproduces the final design values to very acceptable accuracy.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS AND RECOMMENDATIONS

From the comparisons given in Appendix D, it can be seen that the preliminary design code does very well in developing the velocity diagrams and the initial blading geometry necessary for a detailed throughflow design and final geometry calculation [Ref. 3 and 4].

Improvements can be made in order to have the design more detailed as well as improve the code's usability. They are as follows:

- Add the ability to do a multi-stage design (use the output of the previous stage as an input to the new stage).
- Add different input screens for specific design cases (e.g., fan or core).
- Add stress limits (hoop and centrifugal) for fan design.
- Add blockage and bleed to the throughflow calculations.
- Draw scale velocity diagrams.
- Draw blades based on code output (i.e., built in geometry package).
- Add parametric analysis for a range of values (e.g., inlet flow angle).
- Add turbine stage design (in parallel).
- Add the ability to open and save data.
- Add the ability to print the user's results.
- Compile the code into a stand-alone executable.

Incorporation of these improvements will make the code a preliminary turbomachine design software package that can be used as inputs to detailed design packages, as well as providing a needed teaching tool for aircraft engine design.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. DESIGN EQUATIONS

A.1 shows the throughflow equations and A.2 shows the interpolation equations for the incidence, camber and deviation angles.

A.1. THROUGHFLOW EQUATIONS

1. Inlet Conditions

Given: $\dot{m}$, P_t, T_t, M_{W1t}, ω, R, γ

Vary: $M_{Z1t}\left(\equiv \beta_{1t}\right)$, α_{1t} $\left(0 \le \alpha_{1t} < \beta_{1t}\right)$

Where: $\beta_{1t} = \cos^{-1}\left(\dfrac{M_{Z1t}}{M_{W1t}}\right)$

$$M_{1t} = \frac{M_{Z1t}}{M_{W1t}}$$

$$X_{1t} = \sqrt{\frac{\frac{\gamma-1}{2}M_{1t}^2}{1+\frac{\gamma-1}{2}M_{1t}^2}}$$

$$X_{Z1t} = X_{1t}\cdot\cos\alpha_{1t}$$

$$X_{\theta 1t} = X_{Z1t}\cdot\tan\alpha_{1t}$$

$$X_{U1t} = X_{\theta 1t} + X_{Z1t}\cdot\tan\beta_{1t}$$

$$\rho_{t1} = \frac{P_{t1}}{R\cdot T_{t1}}$$

$$V_{t1} = \sqrt{2\cdot C_P\cdot g\cdot T_{t1}}\text{, where } C_P = \left(\frac{\gamma}{\gamma-1}\right)\cdot R$$

$$A_1 = \left(\frac{\dot{m}}{\rho_{t1}\cdot V_{t1}}\right)\cdot\frac{1}{\Phi_{1t}\cdot\cos\alpha_{1t}}\text{, where } \Phi_{1t} = X_{1t}\left(1-X_{1t}^2\right)^{\frac{1}{\gamma-1}}$$

$$r_{1t} = \frac{X_{U1t}\cdot V_{t1}}{\omega}$$

$$r_{1h} = \sqrt{r_{1t}^2 - \frac{A_1}{\pi}}$$

$$r_{ht1} = \frac{r_{1h}}{r_{1t}}$$

$$r_{1m} = \frac{r_{1t}+r_{1h}}{2}$$

Due to Radial Equilibrium $\begin{cases} X_{\theta 1m} = \dfrac{r_{1t}}{r_{1m}} \cdot X_{\theta 1t} \\ X_{Z1m} = X_{Z1t} \end{cases}$

$$\therefore\ \alpha_{1m} = \tan^{-1}\left(\frac{X_{\theta 1m}}{X_{Z1m}}\right) = \tan^{-1}\left[\frac{r_{1t}}{r_{1m}} \cdot \tan\alpha_{1t}\right]$$

$X_{U1m} = \dfrac{r_{1m}}{r_{1t}} \cdot X_{U1t}$ for constant ω

$$\therefore\ \beta_{1m} = \tan^{-1}\left(\frac{X_{U1m} - X_{\theta 1m}}{X_{Z1m}}\right)$$

$$X_{1m} = \frac{X_{Z1m}}{\cos\alpha_{1m}},\ M_{1m} = \sqrt{\frac{\frac{2}{\gamma-1} \cdot X_{1m}^2}{1 - X_{1m}^2}}$$

$$X_{W1m} = \frac{X_{Z1m}}{\cos\beta_{1m}},\ M_{W1m} = \sqrt{\frac{\frac{2}{\gamma-1} \cdot X_{1m}^2}{1 - X_{1m}^2}}$$

2. Rotor Across the Mean Line

Assume: D_{2m}, σ_{2m}, $R_{21} \equiv \dfrac{r_{2m}}{r_{1m}}$ (pitch of mean line), $\phi_{21} \equiv \dfrac{V_{Z2}}{V_{Z1}}$ (change in axial velocity)

Where: $D_{2m} = 1 - \phi_{21} \cdot \dfrac{\cos\beta_{1m}}{\cos\beta_{2m}} + \dfrac{\left(\tan\beta_{1m} - R_{21} \cdot \phi_{21} \cdot \tan\beta_{2m}\right) \cdot \cos\beta_{1m}}{\left(1 + R_{21}\right) \cdot \sigma_{2m}}$

Solve for β_{2m}

$$X_{U2m} = R_{21} \cdot X_{U1m}$$

$\phi_{2m} = \phi_{21} \cdot \phi_{1m} \cdot \left(\dfrac{1}{R_{21}}\right)$, where $\phi_{1m} = \dfrac{X_{Z1m}}{X_{U1m}}$

$$X_{Z2m} = \phi_{2m} \cdot X_{U2m}$$

$$X_{\theta 2m} = X_{U2m} - X_{Z2m} \cdot \tan\beta_{2m}$$

$$\alpha_{2m} = \tan^{-1}\left(\frac{X_{\theta 2m}}{X_{Z2m}}\right)$$

$$r_{sm} = \left[1 - \frac{1}{2}\left(\frac{X_{\theta 1m}}{X_{U1m}} + \frac{X_{\theta 2m}}{X_{U2m}}\right)\right]$$

$$\tau = 1 + 2\left[X_{U2m} \cdot X_{\theta 2m} - X_{U1m} \cdot X_{\theta 1m}\right]$$

$$X_{2m} = \frac{X_{Z2m}}{\cos\alpha_{2m}}$$

$$X_{W2m} = \frac{X_{Z2m}}{\cos\beta_{2m}}$$

$$Y_{2m} = \frac{X_{2m}}{\sqrt{\tau}}$$

$$Y_{W2m} = \frac{X_{W2m}}{\sqrt{\tau}}$$

$$M_{2m} = \sqrt{\frac{\frac{2}{\gamma-1}\cdot Y_{2m}^2}{1-Y_{2m}^2}}$$

$$M_{W2m} = \sqrt{\frac{\frac{2}{\gamma-1}\cdot Y_{W2m}^2}{1-Y_{W2m}^2}}$$

3. Stator Across the Mean Line

Assume: σ_{3m} , $R_{32} = \frac{R_{3m}}{R_{2m}}$, $\phi_{32} = \frac{V_{Z3}}{V_{Z2}}$

Set $\alpha_{3m} = A_{31}\cdot\alpha_{1m}$ (this allows for non-repeating stage calculations)

Then: $D_{3m} = 1-\phi_{32}\cdot\frac{\cos\alpha_{2m}}{\cos\alpha_{3m}} + \frac{\left(\tan\alpha_{2m} - R_{32}\cdot\phi_{32}\cdot\tan\alpha_{3m}\right)\cdot\cos\alpha_{2m}}{\left(1+R_{32}\right)\cdot\sigma_{3m}}$

$$X_{Z3m} = \phi_{32}\cdot X_{Z2m}$$

$$X_{\theta 3m} = X_{Z3m}\cdot\tan\alpha_{3m}$$

$$X_{U3m} = X_{U2m}\cdot R_{32}$$

$$\beta_{3m} = \tan^{-1}\left(\frac{X_{U3m} - X_{\theta 3m}}{X_{Z3m}}\right)$$

$$X_{3m} = \frac{X_{Z3m}}{\cos\alpha_{3m}}$$

$$X_{W3m} = \frac{X_{Z3m}}{\cos\beta_{3m}}$$

$$Y_{3m} = \frac{X_{3m}}{\sqrt{\tau}}$$

$$Y_{W3m} = \frac{X_{W3m}}{\sqrt{\tau}}$$

$$M_{3m} = \sqrt{\frac{\frac{2}{\gamma-1}\cdot Y_{3m}^2}{1-Y_{3m}^2}}$$

$$M_{W3m} = \sqrt{\frac{\frac{2}{\gamma-1}\cdot Y_{W3m}^2}{1-Y_{W3m}^2}}$$

4. Rotor Performance

Initially set $\tilde{\omega}_{SFTC_2}, \tilde{\omega}_{S_2} = 0$

On subsequent iterations $\tilde{\omega}_{SFTC_2}$ and $\tilde{\omega}_{S_2}$ are

$$\beta_\infty = \tan^{-1}\left[\left(\frac{\tan\beta_{1m} + \tan\beta_{2m}}{2}\right)\right]$$

$$C_L = \frac{2}{\sigma_{2m}}\cdot\left(\tan\beta_{1m} - \tan\beta_{2m}\right)\cdot\cos\beta_\infty$$

$$C_{Di} = \frac{1}{4}\cdot C_L^2\cdot\sigma_{2m}\cdot\left(\frac{\delta_2}{H_2}\right)\cdot\frac{1}{\cos\beta_\infty} + 0.04\cdot C_L^2\cdot\sigma_{2m}\cdot\left(\frac{S_2}{H_2}\right)$$

$$\tilde{\omega}_{SFTC_2} = C_{Di}\cdot\frac{\cos^2\beta_{1m}}{\cos^3\beta_\infty}\cdot\sigma_{2m}$$

$$y^* = \frac{1}{4\cdot\gamma\cdot M_{W1m}^2}\left[(\gamma+1)\cdot M_{W1m}^2 - (3-\gamma) + \sqrt{(\gamma+1)\cdot\left\{(\gamma+1)\cdot M_{W1m}^2 - 2\cdot(3-\gamma)\cdot M_{W1m}^2 + \gamma + 9\right\}}\right]$$

$$\frac{P_{te}}{P_{ti}} = \left[\frac{\gamma+1}{2\cdot\gamma\cdot M_{W1m}^2\cdot y^* - (\gamma-1)}\right]^{\frac{1}{\gamma-1}}\cdot\left[\frac{(\gamma+1)\cdot M_{W1m}^2\cdot y^*}{2+(\gamma-1)\cdot M_{W1m}^2\cdot y^*}\right]^{\frac{\gamma}{\gamma-1}}$$

$$\tilde{\omega}_{S_2} = \frac{P_{Ri} - P_{Re}}{P_{Ri} - P_i} = \frac{1 - P_{Re}/P_{Ri}}{1 - P_i/P_{Ri}} = \frac{1 - P_{te}/P_{ti}}{1 - \left[1 + \frac{\gamma-1}{2}\cdot M_{W1m}^2\right]^{-\frac{\gamma}{\gamma-1}}}$$

$$\tilde{\omega}_{P_2} = 2\cdot\sigma_{2m}\cdot\frac{\cos^2\beta_{1m}}{\cos^3\beta_{2m}}\cdot\left[0.005 + 0.16\cdot D_{2m}^4\right]$$

$$\tilde{\omega}_{T_2} = \tilde{\omega}_{P_2} + \tilde{\omega}_{SFTC_2} + \tilde{\omega}_{S_2}$$

$$\frac{T_1}{T_{t1}} = 1 - X_{1m}^2$$

$$\frac{P_1}{P_{t1}} = \left(\frac{T_1}{T_{t1}}\right)^{\frac{\gamma}{\gamma-1}}$$

$$\frac{T_{R1}}{T_{t1}} = \left(\frac{T_1}{T_{t1}}\right) + X_{W1m}^2$$

$$\frac{P_{R1}}{P_{t1}} = \left(\frac{T_{R1}}{T_{t1}}\right)^{\frac{\gamma}{\gamma-1}}$$

$$\frac{T_{E1}}{T_{t1}} = \left(\frac{T_{R1}}{T_{t1}}\right) + X_{U2m}^{2} - X_{U1m}^{2}$$

$$\frac{P_{E1}}{P_{t1}} = \left(\frac{T_{E1}}{T_{t1}}\right)^{\frac{\gamma}{\gamma-1}}$$

$$\frac{P_{E2}}{P_{t1}} = \frac{P_{E1}}{P_{t1}} - \tilde{\omega}_{T_2} \cdot \left[\frac{P_{R1}}{P_{t1}} - \frac{P_1}{P_{t1}}\right]$$

$$\frac{T_{t2}}{T_{t1}} = \tau$$

$$\frac{P_{t2}}{P_{t1}} = \frac{P_{E1}}{P_{t1}} \cdot \left(\frac{T_{t2}}{T_{E2}}\right)^{\frac{\gamma}{\gamma-1}} = \frac{P_{E2}}{P_{t1}} \cdot \left[\frac{\tau}{T_{E2}/T_{t1}}\right]^{\frac{\gamma}{\gamma-1}}, \text{ where } \frac{T_{E2}}{T_{t1}} = \frac{T_{E1}}{T_{t1}}$$

$$\frac{T_2}{T_{t1}} = \tau - X_{2m}^{2}$$

$$\frac{P_2}{P_{t1}} = \frac{P_{t2}}{P_{t1}} \cdot \left(\frac{T_2}{T_{t1}}\right)^{\frac{\gamma}{\gamma-1}} = \frac{P_{t2}}{P_{t1}} \cdot \left(\frac{T_2/T_{t1}}{\tau}\right)^{\frac{\gamma}{\gamma-1}}$$

a. Rotor Annulus

$$\Phi_{2m} = Y_{2m} \cdot \left(1 - Y_{2m}^{2}\right)^{\frac{1}{\gamma-1}}$$

$$\frac{A_2}{A_1} = \frac{\Phi_{1t} \cdot \cos\alpha_{1t}}{\Phi_{2m} \cdot \cos\alpha_{2m}} \cdot \frac{\sqrt{\tau}}{\left(P_{t2}/P_{t1}\right)}$$

$$A_2 = \frac{A_2}{A_1} \cdot A_1$$

$$r_{2m} = r_{1m} \cdot R_{21}$$

$$H_2 = \frac{A_2}{2 \cdot \pi \cdot r_{2m}}$$

$$r_{ht2} = \frac{1 - H_2/2 \cdot r_{2m}}{1 + H_2/2 \cdot r_{2m}}$$

$$r_{t2} = \left(\frac{2}{1 + r_{ht2}}\right) \cdot r_{2m}$$

$$r_{h2} = r_{ht2} \cdot r_{t2}$$

5. Stator Performance

Initially set $\tilde{\omega}_{SFTC_3}, \tilde{\omega}_{S_3} = 0$

On subsequent iterations $\tilde{\omega}_{SFTC_3}$ and $\tilde{\omega}_{S_3}$ are

$$\beta_\infty = \tan^{-1}\left[\left(\frac{\tan\alpha_{2m} + \tan\alpha_{3m}}{2}\right)\right]$$

$$C_L = \frac{2}{\sigma_{3m}} \cdot \left(\tan\alpha_{2m} - \tan\alpha_{3m}\right) \cdot \cos\beta_\infty$$

$$C_{Di} = \frac{1}{4} \cdot C_L^2 \cdot \sigma_{3m} \cdot \left(\frac{\delta_3}{H_3}\right) \cdot \frac{1}{\cos\beta_\infty} + 0.04 \cdot C_L^2 \cdot \sigma_{3m} \cdot \left(\frac{S_3}{H_3}\right)$$

$$\tilde{\omega}_{SFTC_3} = C_{Di} \cdot \frac{\cos^2\alpha_{2m}}{\cos^3\beta_\infty} \cdot \sigma_{3m}$$

$$y^* = \frac{1}{4 \cdot \gamma \cdot M_{2m}^2}\left[(\gamma+1) \cdot M_{2m}^2 - (3-\gamma) + \sqrt{(\gamma+1) \cdot \left\{(\gamma+1) \cdot M_{2m}^2 - 2 \cdot (3-\gamma) \cdot M_{2m}^2 + \gamma + 9\right\}}\right]$$

$$\frac{P_{te}}{P_{ti}} = \left[\frac{\gamma+1}{2 \cdot \gamma \cdot M_{2m}^2 \cdot y^* - (\gamma-1)}\right]^{\frac{1}{\gamma-1}} \cdot \left[\frac{(\gamma+1) \cdot M_{2m}^2 \cdot y^*}{2 + (\gamma-1) \cdot M_{2m}^2 \cdot y^*}\right]^{\frac{\gamma}{\gamma-1}}$$

$$\tilde{\omega}_{S_2} = \frac{P_{ti} - P_{te}}{P_{ti} - P_i} = \frac{1 - P_{te}/P_{ti}}{1 - P_i/P_{ti}} = \frac{1 - P_{te}/P_{ti}}{1 - \left[1 + \frac{\gamma-1}{2} \cdot M_{2m}^2\right]^{-\frac{\gamma}{\gamma-1}}}$$

$$\tilde{\omega}_{P_3} = 2 \cdot \sigma_{3m} \cdot \frac{\cos^2\alpha_{2m}}{\cos^3\alpha_{3m}} \cdot \left[0.005 + 0.16 \cdot D_{3m}^4\right]$$

$$\tilde{\omega}_{T_3} = \tilde{\omega}_{P_3} + \tilde{\omega}_{SFTC_3} + \tilde{\omega}_{S_3}$$

$$\frac{P_{t3}}{P_{t1}} = \frac{P_{t2}}{P_{t1}} - \tilde{\omega}_{T_3} \cdot \left(\frac{P_{t2}}{P_{t1}} - \frac{P_2}{P_{t1}}\right)$$

$$\frac{T_{t3}}{T_{t1}} = \tau$$

$$\frac{T_3}{T_{t1}} = \tau - X_{3m}^2$$

$$\frac{P_3}{P_{t1}} = \left(\frac{P_3}{P_{t3}}\right) \cdot \left(\frac{P_{t3}}{P_{t1}}\right) = \left(\frac{T_3}{T_{t1}}\right)^{\frac{\gamma}{\gamma-1}} \cdot \left(\frac{P_{t3}}{P_{t1}}\right) = \left[\left(\frac{T_3}{T_{t1}}\right) \cdot \left(\frac{T_{t1}}{T_{t3}}\right)\right]^{\frac{\gamma}{\gamma-1}} \cdot \frac{P_{t3}}{P_{t1}} = \left[\frac{T_3/T_{t1}}{\tau}\right]^{\frac{\gamma}{\gamma-1}} \cdot \frac{P_{t3}}{P_{t1}}$$

a. Stator Annulus

$$\Phi_{3m} = Y_{3m} \cdot \left(1 - Y_{3m}^2\right)^{\frac{1}{\gamma-1}}$$

$$\frac{A_3}{A_1} = \frac{\Phi_{1t} \cdot \cos\alpha_{1t}}{\Phi_{3m} \cdot \cos\alpha_{3m}} \cdot \frac{\sqrt{\tau}}{\left(P_{t3} / P_{t1} \right)}$$

$$A_3 = \frac{A_3}{A_1} \cdot A_1$$

$$H_3 = \frac{A_3}{2 \cdot \pi \cdot r_{3m}}$$

$$r_{3m} = r_{2m} \cdot R_{32}$$

$$r_{ht3} = \frac{1 - H_3 / 2 \cdot r_{3m}}{1 + H_3 / 2 \cdot r_{3m}}$$

$$r_{3t} = \left(\frac{2}{1 + r_{ht3}} \right)$$

$$r_{3h} = r_{3m} \cdot r_{3t}$$

6. Rotor & Stator at the Hub and Tip

The below equations are generic forms of the specific equation. The brackets [] replace the rotor (2) and stator (3) subscripts. Also, the parentheses () replace the hub (1) and tip (3) subscripts.

$$X_{\theta[]()} = X_{\theta[]m} \cdot \frac{r_{[]m}}{r_{[]()}}$$

$$X_{U[]()} = X_{U[]m} \cdot \frac{r_{[]()}}{r_{[]m}}$$

$$\alpha_{[]()} = \tan^{-1} \frac{X_{\theta[]()}}{X_{Z[]m}}$$

$$\beta_{[]()} = \tan^{-1} \frac{X_{U[]()} - X_{\theta[]()}}{X_{Z[]m}}$$

$$X_{[]()} = \frac{X_{Z[]m}}{\cos\alpha_{[]()}}$$

$$X_{W[]()} = \frac{X_{Z[]m}}{\cos\beta_{[]()}}$$

$$Y_{[]()} = \frac{X_{[]()}}{\sqrt{\tau}}$$

$$Y_{W[]()} = \frac{X_{W[]()}}{\sqrt{\tau}}$$

$$M_{[]0} = \sqrt{\frac{\frac{2}{\gamma-1}\cdot Y_{[]0}^2}{1-Y_{[]0}^2}}$$

$$M_{W[]0} = \sqrt{\frac{\frac{2}{\gamma-1}\cdot Y_{W[]0}^2}{1-Y_{W[]0}^2}}$$

7. Blade Geometry

The below equations are generic forms of the specific equation. The brackets [] replace the rotor (2) and stator (3) subscripts.

$$H_{[]} = r_{[]t} - r_{[]h}$$

$$C_{[]}^{'} = \frac{H_{[]}}{AR}$$

$$Z_{[]}^{'} = \frac{2\cdot\pi\cdot r_{[]m}}{C_{[]}^{'}}\cdot\sigma_{[]m}$$

$Z_{[]}$ is chosen by the user at this point.

$$AR_{Rev} = \frac{H_{[]}}{\left(2\cdot\pi\cdot r_{[]m}\cdot\sigma_{[]m}/Z_{[]}\right)}$$

$$C_{Rev[]} = \frac{H_{[]}}{AR_{Rev}}$$

$$S_{[]} = \frac{C_{Rev[]}}{\sigma_{[]m}}$$

8. Efficiency

$$\Pi_C = \frac{P_{t3}}{P_{t1}}$$

$$\tau_C = \frac{T_{t2}}{T_{t1}}$$

$$\eta_C = \frac{\Pi_C^{\frac{\gamma}{\gamma-1}}-1}{\tau_C - 1}$$

A.2. INTERPOLATION EQUATIONS

Starting with a general quadratic equation

$$D^i(R) = A_i\cdot R^2 + B_i\cdot R + C_i$$

solve for A_i, B_i, and C_i using known points i-1, i, and i+1.

$$A_i = \left(\frac{1}{R_{i+1} - R_{i-1}} \right) \cdot \left[\left(\frac{D_{i+1} - D_i}{R_{i+1} - R_i} \right) - \left(\frac{D_i - D_{i-1}}{R_i - R_{i-1}} \right) \right]$$

$$B_i = \left(\frac{D_i - D_{i-1}}{R_i - R_{i-1}} \right) - A_i \cdot \left(R_i + R_{i-1} \right)$$

$$C_i = D_i - A_i \cdot R_i^2 - B_i \cdot R_i$$

Over the first and last intervals, only one quadratic can be defined, so that

$$D_{int}\left(1\right) = \frac{A_1}{3} \cdot \left(R_2^3 - R_1^2 \right) + \frac{B_2}{2} \cdot \left(R_2^2 - R_1^2 \right) + C_2 \cdot \left(R_2 - R_1 \right)$$

$$D_{int}\left(N\right) = \frac{A_N}{3} \cdot \left(R_{N+1}^3 - R_N^3 \right) + \frac{B_N}{2} \cdot \left(R_{N+1}^2 - R_N^2 \right) + C_N \cdot \left(R_{N+1} - R_N \right)$$

The complete integral is given by

$$\int_{R_i}^{R_{N+1}} \bar{D}\left(R\right) dr = D_{int}\left(1\right) + \sum_{i=2}^{N-1} D_{int}\left(i\right) + D_{int}\left(N\right)$$

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. PROGRAM SOURCE DATA

B.1. SCREEN SNAPSHOTS

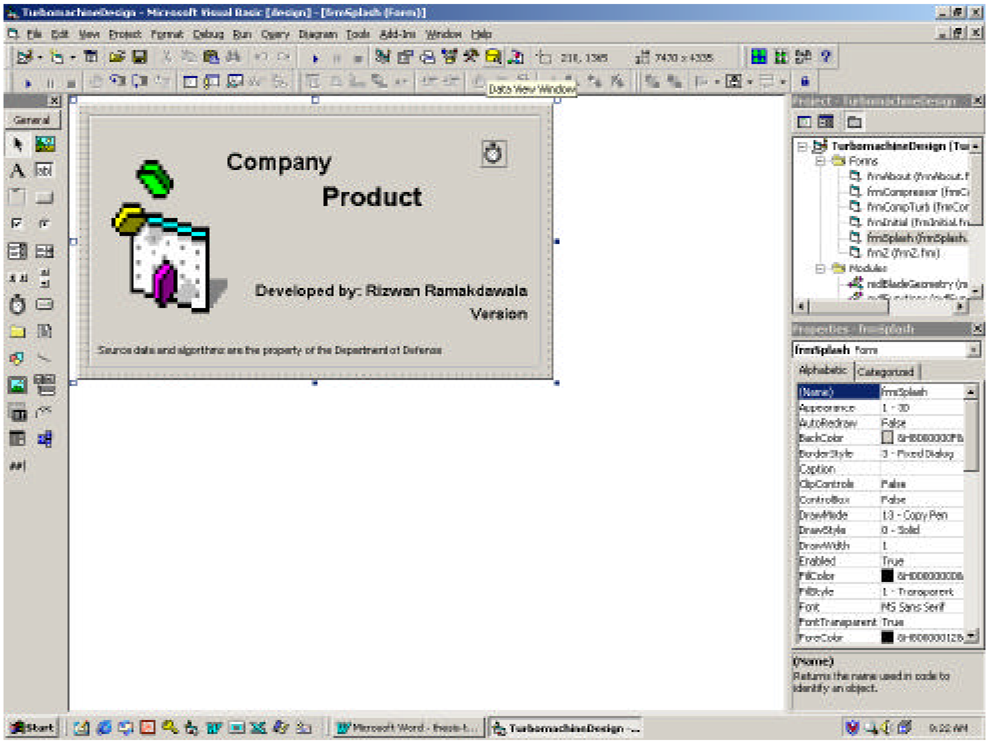

Figure B1. Splash Screen

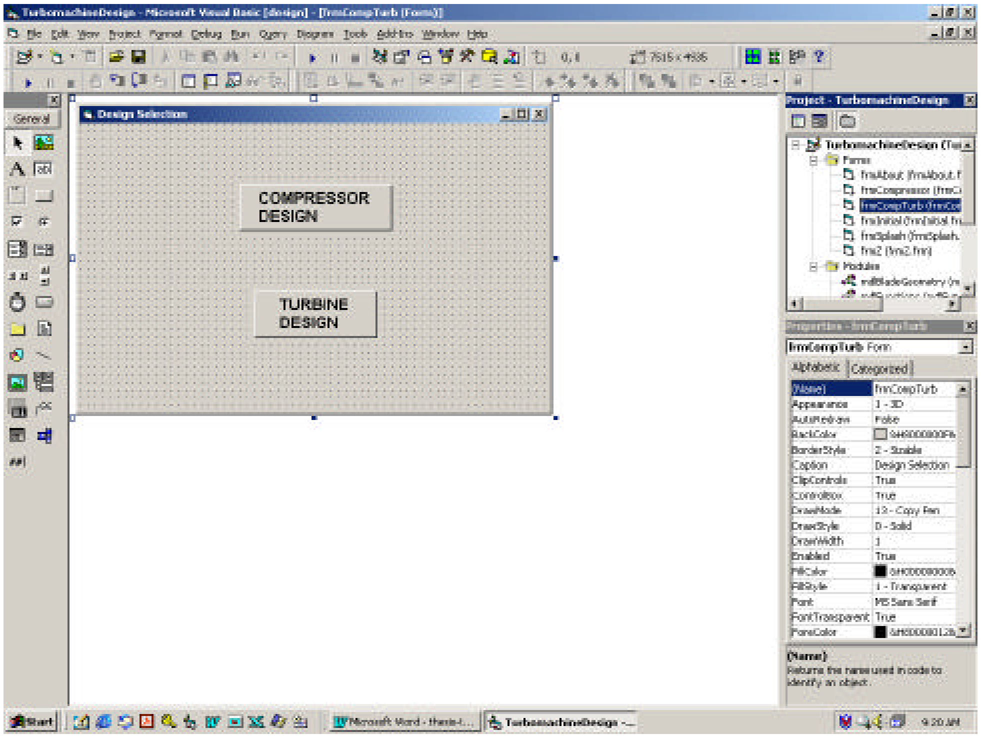

Figure B2. Compressor/Turbine Selection Screen

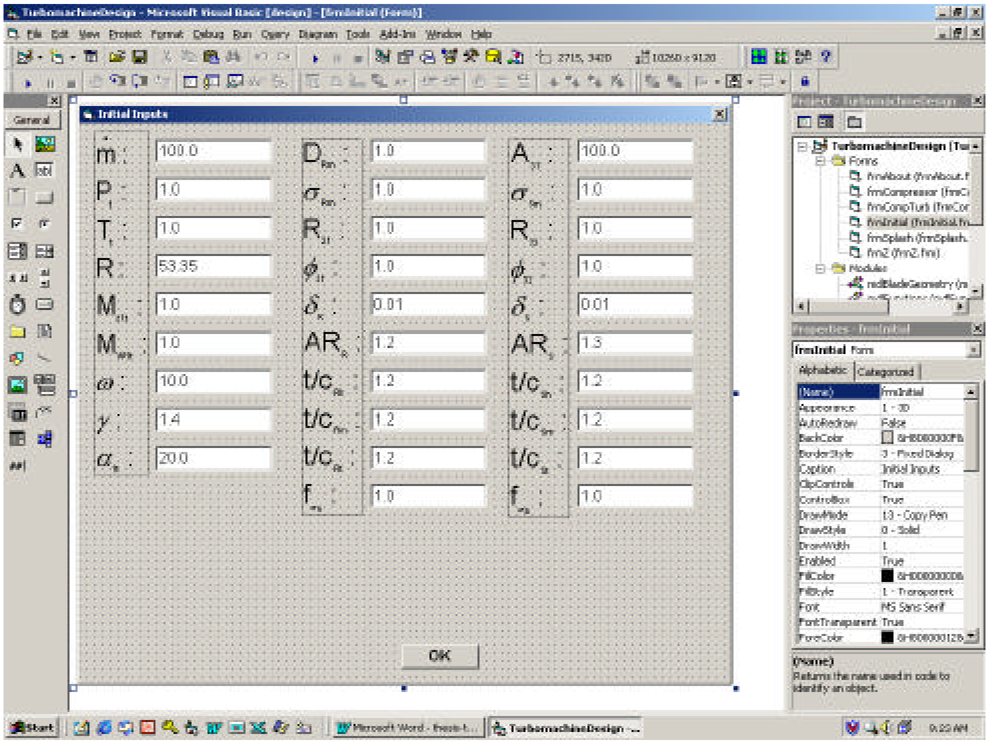

Figure B3. Initial Inputs Screen

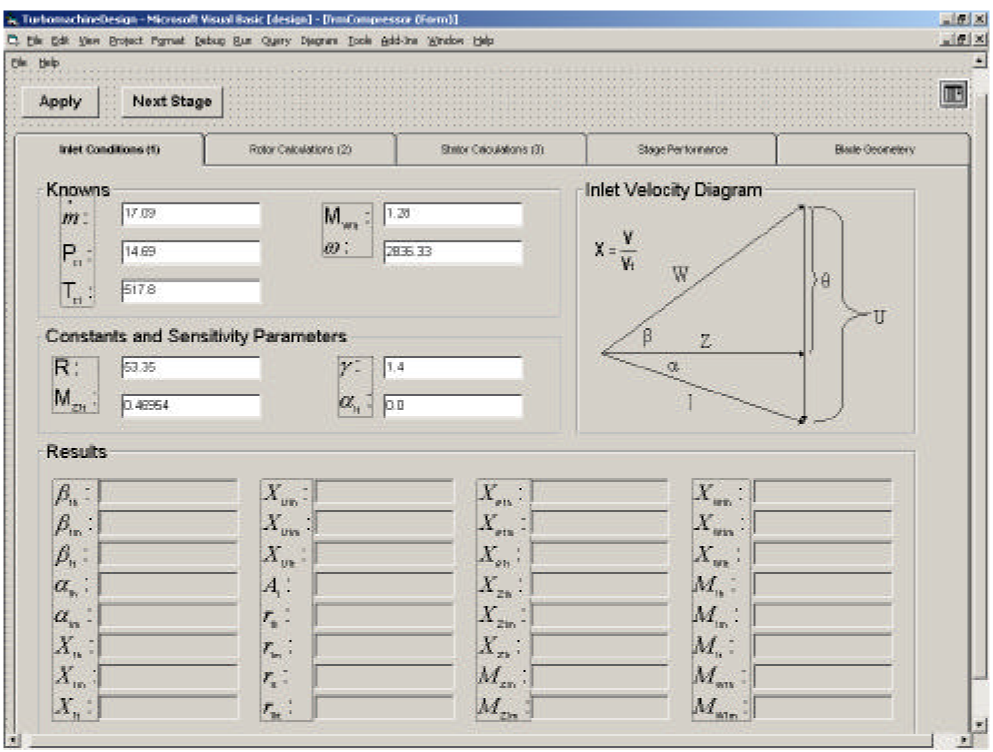

Figure B4. Compressor Design Screen: Inlet Conditions (1)

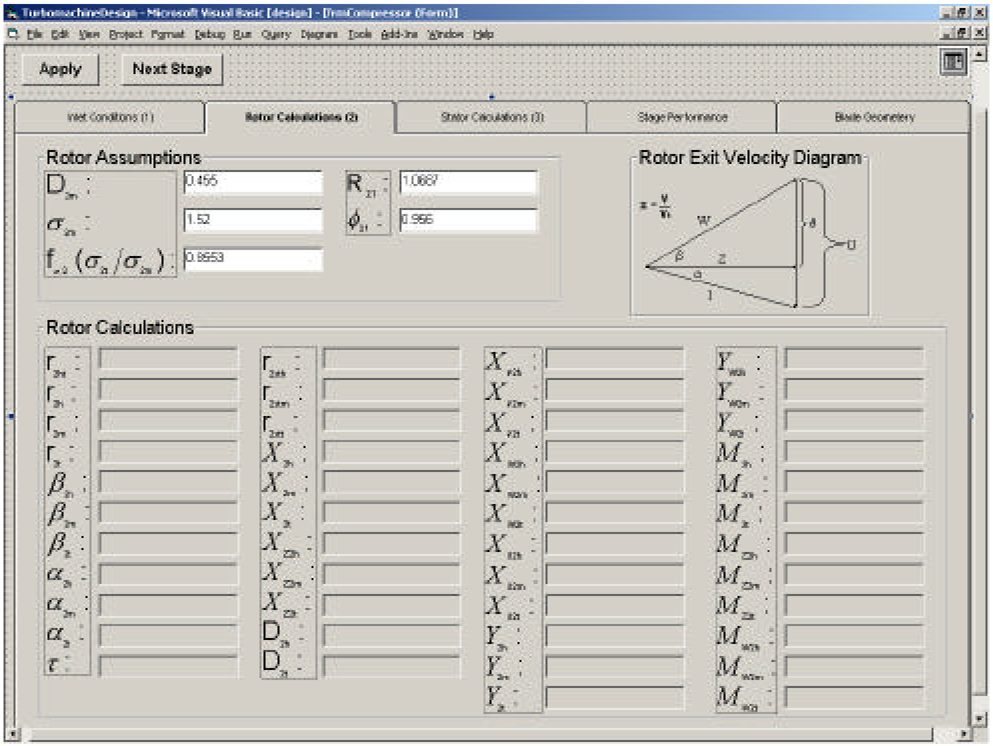

Figure B5. Compressor Design Screen: Rotor Calculations (2)

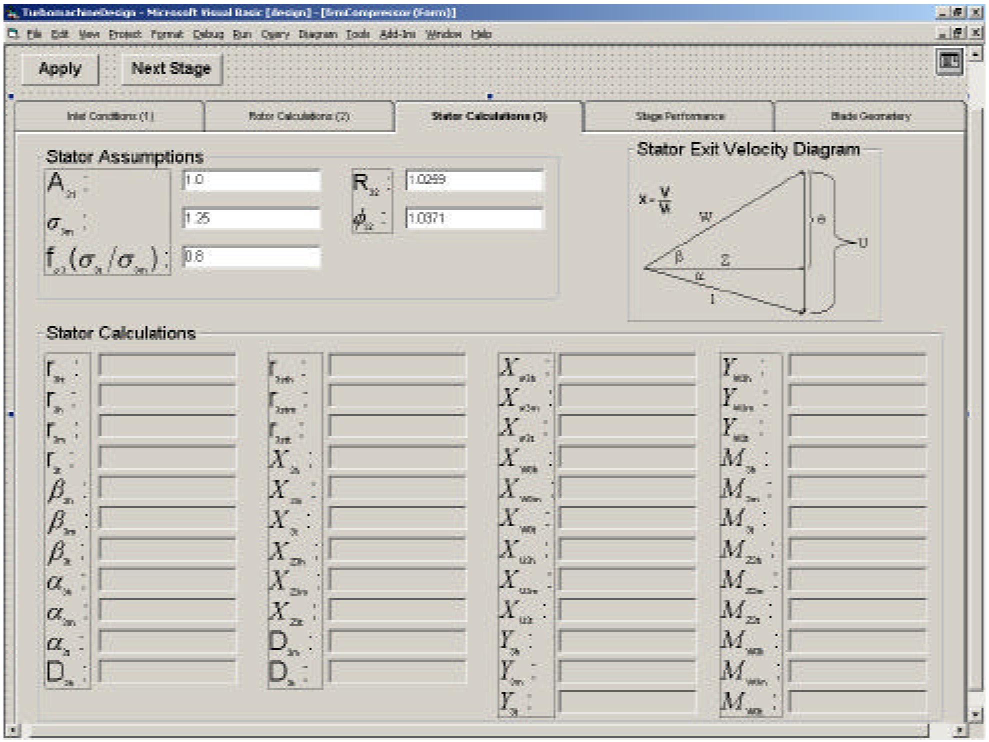

Figure B6. Compressor Design Screen: Stator Calculations (3)

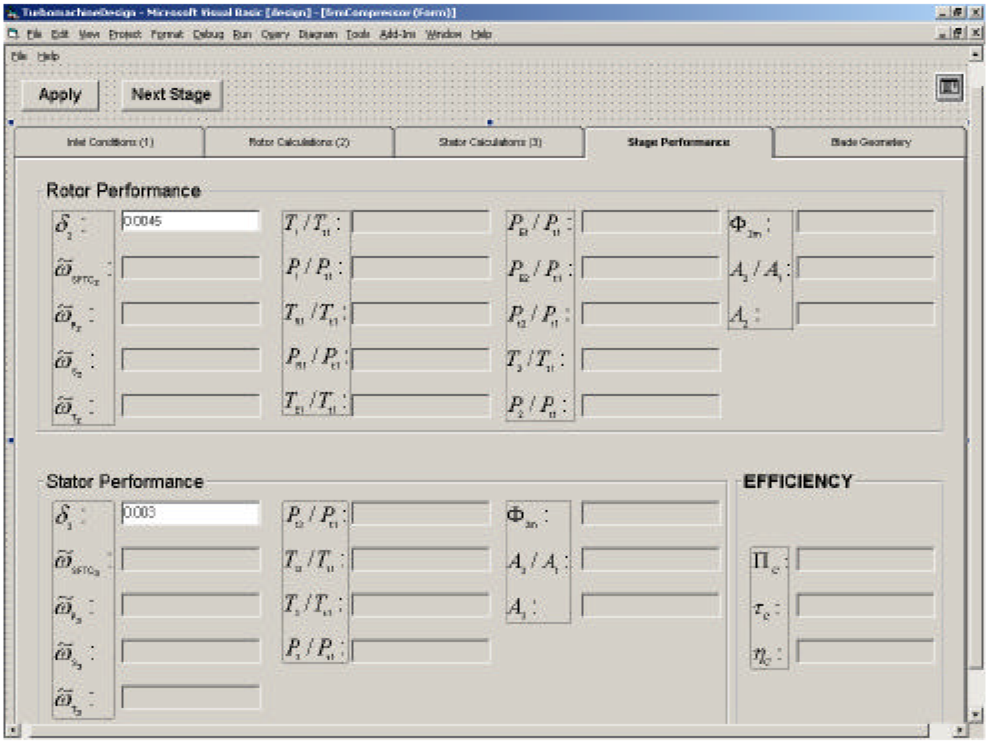

Figure B7. Compressor Design Screen: Stage Performance

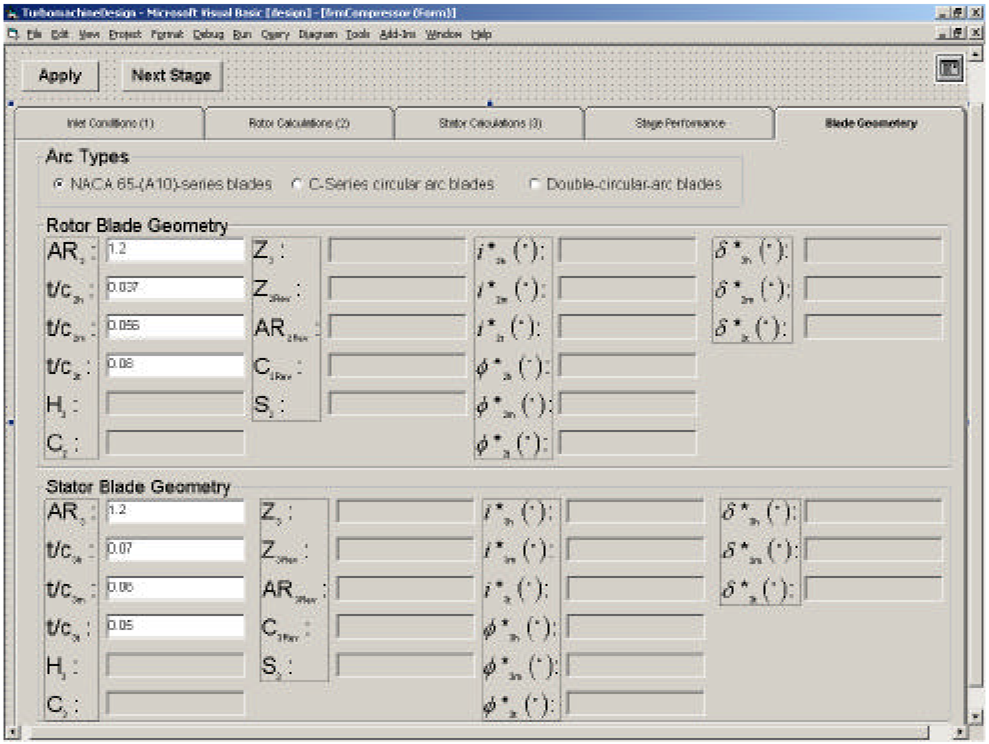

Figure B8. Compressor Design Screen: Blade Geometry

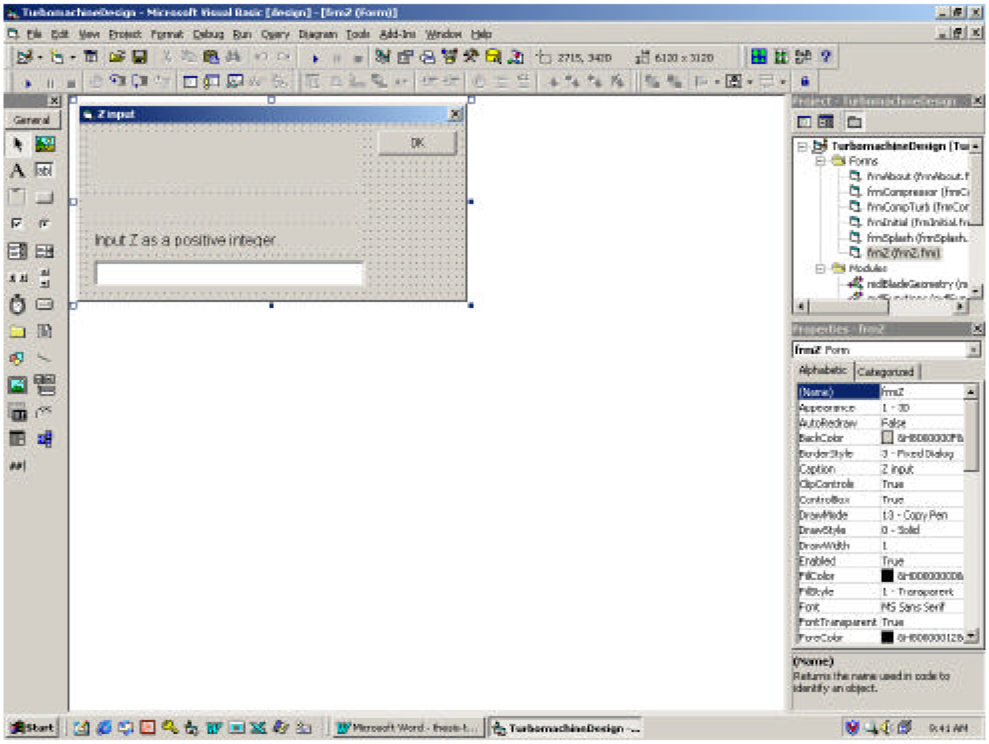

Figure B9. Blade Number Input Screen

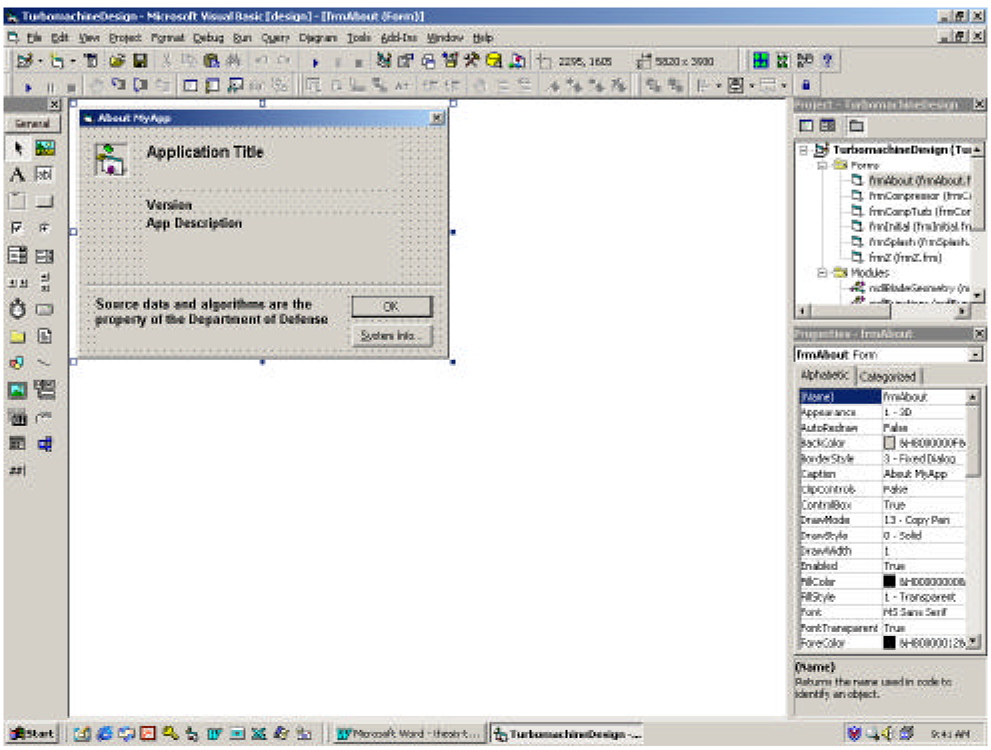

Figure B10. About Screen

B.2. KEY VARIABLES

Error! Not a valid link.

Table B1. Velocity Diagram Variables

Error! Not a valid link.

Table B2. Stage Performance Variables

Error! Not a valid link.

Table B3. Blade Geometry Variables

B.3. SOURCE CODE

Lines with an apostrophe (') in the front is a comment line in the code and is not an executed statement.

1. Splash Screen Code

```
Option Explicit
Private Sub Form_Load()
        lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." & App.Revision
        lblProductName.Caption = App.Title
        lblCompany.Caption = App.CompanyName
End Sub
Private Sub tmrSplash_Timer()
        Unload Me
        frmCompTurb.Show
End Sub
```

2. Compressor/Turbine Selection Screen Code

```
Private Sub cmdComp_Click()
        Unload Me
        frmInitial.Show
End Sub
```

3. Initial Input Screen Code

```
Option Explicit
Option Base 1
Private Sub OKButton_Click()
        frmInitial.Hide
```

```
    frmCompressor.Show
    ' Transfer initial values to the main form.
End Sub
```

4. Compressor Design Screen Code

```
Option Explicit
Option Base 1
Public i As Integer
' Integer to use as an array counter from inlet (i = 1) to rotor (i=2) to stator (i=3)
' It should be noted that although some arrays have no inlet values (e.g. diffusion and deg. of
    reation)
' we still set the array value to 3 versus 2.  This is to keep consistent with the numbering.
' arrays that don't have the i=1 value assigned are null and irrelevent since they are not used
' in any calculations.
Public j As Integer
' integer to use as an array counter from hub (j = 1) to mean (j = 2) to tip (j = 3)
' It should be noted that although hub, mn and tip have been set up a constants
' j will be used to calculate the minimum loss incidence angle and the camber angle


Public ksh As Single
' correction factor for shape
Public slopegraph As Boolean
' checks which graph should be used to calculate the slope factor
Const testvar As Single = 0.0001
' exit criteria for main loop
Public loopcount As Integer
' loop counter to track the number of iterations
Dim tempvar(3) As Double
' holder for previous loss value for comparison
Public CompEff As Double
' compressor efficiency
Private Sub cmdApply_Click()
    Sequence
    UpdateVelDiag
    UpdatePerf
    UpdateGeo
End Sub
```

```
Private Sub Form_Initialize()
        frmCompressor.txtmdot.Text = frmInitial.txtmdot.Text
        frmCompressor.txtPt1.Text = frmInitial.txtPt1.Text
        frmCompressor.txtTt1.Text = frmInitial.txtTt1.Text
        frmCompressor.txtR.Text = frmInitial.txtR.Text
        frmCompressor.txtMzt(1).Text = frmInitial.txtMz1t.Text
        frmCompressor.txtMwt(1).Text = frmInitial.txtMw1t.Text
        frmCompressor.txtspeed.Text = frmInitial.txtomega.Text
        frmCompressor.txtgamma.Text = frmInitial.txtgamma.Text
        frmCompressor.txtalphat(1).Text = frmInitial.txtAlpha1t.Text
        frmCompressor.txtDm(2).Text = frmInitial.txtDRm.Text
        frmCompressor.txtsigmam(2).Text = frmInitial.txtsigmaRm.Text
        frmCompressor.txtsigmam(3).Text = frmInitial.txtsigmaSm.Text
        frmCompressor.txtR21.Text = frmInitial.txtR21.Text
        frmCompressor.txtR32.Text = frmInitial.txtR32.Text
        frmCompressor.txtphi21.Text = frmInitial.txtphi21.Text
        frmCompressor.txtphi32.Text = frmInitial.txtphi32.Text
        frmCompressor.txtA31.Text = frmInitial.txtA31.Text
        frmCompressor.txtDelta(2).Text = frmInitial.txtDeltar.Text
        frmCompressor.txtDelta(3).Text = frmInitial.txtDeltas.Text
        frmCompressor.txtAR(2).Text = frmInitial.txtARr.Text
        frmCompressor.txtAR(3).Text = frmInitial.txtARs.Text
        frmCompressor.txttch(2).Text = frmInitial.txttcrh.Text
        frmCompressor.txttch(3).Text = frmInitial.txttcsh.Text
        frmCompressor.txttcm(2).Text = frmInitial.txttcrm.Text
        frmCompressor.txttcm(3).Text = frmInitial.txttcsm.Text
        frmCompressor.txttct(2).Text = frmInitial.txttcrt.Text
        frmCompressor.txttct(3).Text = frmInitial.txttcst.Text
        frmCompressor.txtfsigma(2).Text = frmInitial.txtfsigmar.Text
        frmCompressor.txtfsigma(3).Text = frmInitial.txtfsigmas.Text
        ' Initialize Inlet Condition Variables.
        alpha(1, 3) = frmCompressor.txtalphat(1).Text
        gamma = frmCompressor.txtgamma.Text
        Mz(1, 3) = frmCompressor.txtMzt(1).Text
        mdot = frmCompressor.txtmdot.Text
        Mw(1, 3) = frmCompressor.txtMwt(1).Text
        omega = frmCompressor.txtspeed.Text
```

```
        Pt(1) = frmCompressor.txtPt1.Text
        Rbar = frmCompressor.txtR.Text
        Tt(1) = frmCompressor.txtTt1.Text
        ' Initialize Rotor Assumption Variables
        D(2, 2) = frmCompressor.txtDm(2).Text
        sigma(2, 2) = frmCompressor.txtsigmam(2).Text
        pitch(2) = frmCompressor.txtR21.Text
        phi21 = frmCompressor.txtphi21.Text
        fsigma(2) = frmCompressor.txtfsigma(2).Text
        ' Initialize Stator Assumption Variables
        A31 = frmCompressor.txtA31.Text
        sigma(3, 2) = frmCompressor.txtsigmam(3).Text
        pitch(3) = frmCompressor.txtR32.Text
        phi32 = frmCompressor.txtphi32.Text
        fsigma(3) = frmCompressor.txtfsigma(3).Text
        ' Initialize Stage Performance variables
        delta(2) = frmCompressor.txtDelta(2).Text
        delta(3) = frmCompressor.txtDelta(3).Text
        ' initialize blade geometry variables
        For i = 2 To 3
                For j = 1 To 3
                        AR(i) = frmCompressor.txtAR(i).Text
                        If j = 1 Then
                                tc(i, j) = frmCompressor.txttch(i).Text
                        ElseIf j = 2 Then
                                tc(i, j) = frmCompressor.txttcm(i).Text
                        ElseIf j = 3 Then
                                tc(i, j) = frmCompressor.txttct(i).Text
                        End If
                Next j
        Next i
End Sub
Private Sub UpdateVelDiag()
        For i = 1 To 3
                For j = 1 To 3
                        If j = 1 Then
                                txtBetah(i).Text = beta(i, j)
```

```
        txtAlphah(i).Text = alpha(i, j)
        txtXh(i).Text = x(i, j)
        txtXUh(i).Text = Xu(i, j)
        txtXzh(i).Text = Xz(i, j)
        txtXthetah(i).Text = Xtheta(i, j)
        txtXwh(i).Text = Xw(i, j)
        txtrh(i).Text = R(i, j)
        txtMh(i).Text = M(i, j)
        txtMwh(i).Text = Mw(i, j)
        txtMzh(i).Text = Mz(i, j)
        If i = 2 Then
            txtrsth(i).Text = rst(i, j)
            txtDh(i).Text = D(i, j)
            txtYh(i).Text = Y(i, j)
            txtYwh(i).Text = Yw(i, j)
        ElseIf i = 3 Then
            txtrsth(i).Text = rst(i, j)
            txtDh(i).Text = D(i, j)
            txtYh(i).Text = Y(i, j)
            txtYwh(i).Text = Yw(i, j)
        End If
    End If
    If j = 2 Then
        txtbetam(i).Text = beta(i, j)
        txtalpham(i).Text = alpha(i, j)
        txtXm(i).Text = x(i, j)
        txtXUm(i).Text = Xu(i, j)
        txtXzm(i).Text = Xz(i, j)
        txtXthetam(i).Text = Xtheta(i, j)
        txtXwm(i).Text = Xw(i, j)
        txtrm(i).Text = R(i, j)
        txtMm(i).Text = M(i, j)
        txtMwm(i).Text = Mw(i, j)
        txtMzm(i).Text = Mz(i, j)
        If i = 2 Then
            txtrstm(i).Text = rst(i, j)
            txtYm(i).Text = Y(i, j)
```

```
                txtYwm(i).Text = Yw(i, j)
        ElseIf i = 3 Then
                txtrstm(i).Text = rst(i, j)
                txtYm(i).Text = Y(i, j)
                txtYwm(i).Text = Yw(i, j)
                txtDm(i) = D(i, j)
        End If
End If
If j = 3 Then
        txtbetat(i).Text = beta(i, j)
        txtXt(i).Text = x(i, j)
        txtXUt(i).Text = Xu(i, j)
        txtXzt(i).Text = Xz(i, j)
        txtXthetat(i).Text = Xtheta(i, j)
        txtXwt(i).Text = Xw(i, j)
        txtrt(i).Text = R(i, j)
        txtMt(i).Text = M(i, j)
        If i = 2 Then
                txtalphat(i).Text = alpha(i, j)
                txtMwt(i).Text = Mw(i, j)
                txtMzt(i).Text = Mz(i, j)
                txtrstt(i).Text = rst(i, j)
                txtDt(i).Text = D(i, j)
                txtYt(i).Text = Y(i, j)
                txtYwt(i).Text = Yw(i, j)
        ElseIf i = 3 Then
                txtalphat(i).Text = alpha(i, j)
                txtMwt(i).Text = Mw(i, j)
                txtMzt(i).Text = Mz(i, j)
                txtrstt(i).Text = rst(i, j)
                txtDt(i).Text = D(i, j)
                txtYt(i).Text = Y(i, j)
                txtYwt(i).Text = Yw(i, j)
        End If
End If
txtA(i).Text = A(i)
txtrht(i).Text = rht(i)
```

```
            Next j
        Next i
        txttau.Text = tau
End Sub
Private Sub UpdatePerf()
        For i = 1 To 3
            txtTTt1(i).Text = TTt1(i)
            txtPPt1(i).Text = PPt1(i)
        Next i
        For i = 1 To 2
            txtPEPt1(i).Text = PEPt1(i)
        Next i
        For i = 2 To 3
            txtDelta(i).Text = delta(i)
            txtomegasftc(i).Text = omegasftc(i)
            txtomegap(i).Text = omegap(i)
            txtomegasl(i).Text = omegas(i)
            txtomega(i).Text = omegat(i)
            txtAA1(i).Text = AA1(i)
            txtPhim(i).Text = Capphi(i, mn)
            txtPtPt1(i).Text = PtPt1(i)
        Next i
        txtTETt1(1).Text = TETt1(1)
        txtTR1Tt1.Text = Tr1Tt1
        txtPR1Pt1.Text = Pr1Pt1
        txtTt3Tt1.Text = TtTt1(3)
        txtPI.Text = PtPt1(3)
        txtBTau.Text = tau
        txteff.Text = CompEff
End Sub
Private Sub UpdateGeo()
        For i = 2 To 3
            txtH(i).Text = h(i)
            txtC(i).Text = C(i)
            txtZ(i).Text = Z(i)
            txtZrev(i).Text = intZ(i)
            txtARrev(i).Text = ARrev(i)
```

```
            txtCrev(i).Text = Crev(i)
            txtS(i).Text = S(i)
            txtcamberh(i) = camber(i, 1)
            txtcamberm(i) = camber(i, 2)
            txtcambert(i) = camber(i, 3)
            txtincidenceh(i) = icor(i, 1)
            txtincidencem(i) = icor(i, 2)
            txtincidencet(i) = icor(i, 3)
            txtdevh(i) = dref(i, 1)
            txtdevm(i) = dref(i, 2)
            txtdevt(i) = dref(i, 3)
        Next i
End Sub
Private Sub Form_Load()
        ' Initialize Inlet Condition Variables.
        alpha(1, 3) = frmCompressor.txtalphat(1).Text
        gamma = frmCompressor.txtgamma.Text
        Mz(1, 3) = frmCompressor.txtMzt(1).Text
        mdot = frmCompressor.txtmdot.Text
        Mw(1, 3) = frmCompressor.txtMwt(1).Text
        omega = frmCompressor.txtspeed.Text
        Pt(1) = frmCompressor.txtPt1.Text
        Rbar = frmCompressor.txtR.Text
        Tt(1) = frmCompressor.txtTt1.Text
        ' Initialize Rotor Assumption Variables
        D(2, 2) = frmCompressor.txtDm(2).Text
        sigma(2, 2) = frmCompressor.txtsigmam(2).Text
        pitch(2) = frmCompressor.txtR21.Text
        phi21 = frmCompressor.txtphi21.Text
        fsigma(2) = frmCompressor.txtfsigma(2).Text
        ' Initialize Stator Assumption Variables
        A31 = frmCompressor.txtA31.Text
        sigma(3, 2) = frmCompressor.txtsigmam(3).Text
        pitch(3) = frmCompressor.txtR32.Text
        phi32 = frmCompressor.txtphi32.Text
        fsigma(3) = frmCompressor.txtfsigma(3).Text
        ' Initialize Stage Performance variables
```

```
        delta(2) = frmCompressor.txtDelta(2).Text
        delta(3) = frmCompressor.txtDelta(3).Text
        ' initialize blade geometry variables
        For i = 2 To 3
                For j = 1 To 3
                        AR(i) = frmCompressor.txtAR(i).Text
                        If j = 1 Then
                                tc(i, j) = frmCompressor.txttch(i).Text
                        ElseIf j = 2 Then
                                tc(i, j) = frmCompressor.txttcm(i).Text
                        ElseIf j = 3 Then
                                tc(i, j) = frmCompressor.txttct(i).Text
                        End If
                Next j
        Next i
End Sub
Private Sub mnuAbout_Click()
        frmAbout.Show
End Sub
Private Sub mnuContents_Click()
        MsgBox "This function is not available yet.", vbInformation, "Not Implemented"
End Sub
Private Sub mnuExit_Click()
        End
End Sub
Private Sub mnuOpen_Click()
        dlgFile.ShowOpen
End Sub
Private Sub mnuPrint_Click()
        dlgFile.ShowPrinter
        Printer.Copies = dlgFile.Copies
        Printer.Orientation = dlgFile.Orientation
        Printer.Print A(i)
        Printer.EndDoc
End Sub
Private Sub mnuSave_Click()
        dlgFile.ShowSave
```

```
End Sub
Private Sub mnuSearch_Click()
    MsgBox "This function is not available yet.", vbInformation, "Not Implemented"
End Sub
Public Sub Sequence()
    i = 1
    InletCond
    HubCalc i
    If optNACA = True Then
        ksh = 1
        slopegraph = True
    End If
    If optC = True Then
        ksh = 1.1
        slopegraph = False
    End If
    If optDCA = True Then
        ksh = 0.7
        slopegraph = False
    End If
    For i = 2 To 3
        MeanCalc i
    Next i
    loopcount = 1
    Do
        For i = 2 To 3
            If loopcount = 1 Then
                omegat(i) = 0
                omegasftc(i) = 0
                omegas(i) = 0
            End If
            tempvar(i) = omegat(i)
            If i = 2 Then
                RotorPerf i, loopcount
            ElseIf i = 3 Then
                StatorPerf i, loopcount
            End If
```

```
                HubTipCalc i
                BladeGeo i, loopcount
            Next i
            loopcount = loopcount + 1
        Loop Until Abs(omegat(2) - tempvar(2)) < testvar And Abs(omegat(3) - tempvar(3)) <
            testvar
        For i = 2 To 3
            For j = 1 To 3
                Incidence i, j, slopegraph, ksh
            Next j
        Next i
        CompEff = ((PtPt1(3) ^ g1g(gamma)) - 1) / (tau - 1)
End Sub
Private Sub txtA31_KeyPress(KeyAscii As Integer)
        Select Case KeyAscii
            Case vbKey0 To vbKey9
            Case vbDecimal, 46
            Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
            Case 45
                If Len(txtA31.Text) <> 0 Then
                    KeyAscii = 0 ' ignore keystroke
                    Beep
                End If
            Case Else
                KeyAscii = 0
                Beep
        End Select
End Sub
Private Sub txtalphat_KeyPress(Index As Integer, KeyAscii As Integer)
        Select Case KeyAscii
            Case vbKey0 To vbKey9
            Case vbDecimal, 46
            Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
            Case 45
                If Len(txtalphat(1).Text) <> 0 Then
                    KeyAscii = 0 ' ignore keystroke
                    Beep
```

```
                End If
            Case Else
                KeyAscii = 0
               Beep
        End Select
End Sub
Private Sub txtAR_KeyPress(Index As Integer, KeyAscii As Integer)
        Select Case KeyAscii
            Case vbKey0 To vbKey9
            Case vbDecimal, 46
            Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
            Case 45
                If Len(t xtAR(2).Text) <> 0 Then
                    KeyAscii = 0 ' ignore keystroke
                    Beep
                ElseIf Len(txtAR(3).Text) <> 0 Then
                    KeyAscii = 0
                    Beep
                End If
            Case Else
                KeyAscii = 0
                Beep
        End Select
End Sub
Private Sub txtDelta_KeyPress(Index As Integer, KeyAscii As Integer)
        Select Case KeyAscii
            Case vbKey0 To vbKey9
            Case vbDecimal, 46
            Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
            Case 45
                If Len(txtDelta(2).Text) <> 0 Then
                    KeyAscii = 0 ' ignore keystroke
                    Beep
                ElseIf Len(txtDelta(3).Text) <> 0 Then
                    KeyAscii = 0
                    Beep
                End If
```

```
                Case Else
                        KeyAscii = 0
                        Beep
        End Select
End Sub
Private Sub txtDm_KeyPress(Index As Integer, KeyAscii As Integer)
        Select Case KeyAscii
                Case vbKey0 To vbKey9
                Case vbDecimal, 46
                Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
                Case 45
                        If Len(txtDm(2).Text) <> 0 Then
                                KeyAscii = 0 ' ignore keystroke
                                Beep
                        End If
                Case Else
                        KeyAscii = 0
                        Beep
        End Select
End Sub
Private Sub txtgamma_KeyPress(KeyAscii As Integer)
        Select Case KeyAscii
                Case vbKey0 To vbKey9
                Case vbDecimal, 46
                Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
                Case 45
                        If Len(txtgamma.Text) <> 0 Then
                                KeyAscii = 0 ' ignore keystroke
                                Beep
                        End If
                Case Else
                        KeyAscii = 0
                        Beep
        End Select
End Sub
Private Sub txtmdot_KeyPress(KeyAscii As Integer)
        Select Case KeyAscii
```

```
        Case vbKey0 To vbKey9
        Case vbDecimal, 46
        Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
        Case 45
            If Len(txtmdot.Text) <> 0 Then
                KeyAscii = 0 ' ignore keystroke
                Beep
            End If
        Case Else
            KeyAscii = 0
            Beep
    End Select
End Sub
Private Sub txtMwt_KeyPress(Index As Integer, KeyAscii As Integer)
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbDecimal, 46
        Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
        Case 45
            If Len(txtMwt(1).Text) <> 0 Then
                KeyAscii = 0 ' ignore keystroke
                Beep
            End If
        Case Else
            KeyAscii = 0
            Beep
    End Select
End Sub
Private Sub txtMzt_KeyPress(Index As Integer, KeyAscii As Integer)
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbDecimal, 46
        Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
        Case 45
            If Len(txtMzt(1).Text) <> 0 Then
                KeyAscii = 0 ' ignore keystroke
                Beep
```

```
                End If
            Case Else
                KeyAscii = 0
                Beep
        End Select
End Sub
Private Sub txtphi21_KeyPress(KeyAscii As Integer)
        Select Case KeyAscii
            Case vbKey0 To vbKey9
            Case vbDecimal, 46
            Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
            Case 45
                If Len(txtphi21.Text) <> 0 Then
                    KeyAscii = 0 ' ignore keystroke
                    Beep
                End If
            Case Else
                KeyAscii = 0
                Beep
        End Select
End Sub
Private Sub txtphi32_KeyPress(KeyAscii As Integer)
        Select Case KeyAscii
            Case vbKey0 To vbKey9
            Case vbDecimal, 46
            Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
            Case 45
                If Len(txtphi32.Text) <> 0 Then
                    KeyAscii = 0 ' ignore keystroke
                    Beep
                End If
            Case Else
                KeyAscii = 0
                Beep
        End Select
End Sub
Private Sub txtPt1_KeyPress(KeyAscii As Integer)
```

```
		Select Case KeyAscii
			Case vbKey0 To vbKey9
			Case vbDecimal, 46
			Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
			Case 45
				If Len(txtPt1.Text) <> 0 Then
					KeyAscii = 0 ' ignore keystroke
					Beep
				End If
			Case Else
				KeyAscii = 0
				Beep
		End Select
	End Sub
	Private Sub txtR_KeyPress(KeyAscii As Integer)
		Select Case KeyAscii
			Case vbKey0 To vbKey9
			Case vbDecimal, 46
			Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
			Case 45
				If Len(txtR.Text) <> 0 Then
					KeyAscii = 0 ' ignore keystroke
					Beep
				End If
			Case Else
				KeyAscii = 0
				Beep
		End Select
	End Sub
	Private Sub txtR21_KeyPress(KeyAscii As Integer)
		Select Case KeyAscii
			Case vbKey0 To vbKey9
			Case vbDecimal, 46
			Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
			Case 45
				If Len(txtR21.Text) <> 0 Then
					KeyAscii = 0 ' ignore keystroke
```

```
                Beep
            End If
        Case Else
            KeyAscii = 0
            Beep
    End Select
End Sub
Private Sub txtR32_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbDecimal, 46
        Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
        Case 45
            If Len(txtR32.Text) <> 0 Then
                KeyAscii = 0 ' ignore keystroke
                Beep
            End If
        Case Else
            KeyAscii = 0
            Beep
    End Select
End Sub
Private Sub txtsigmam_KeyPress(Index As Integer, KeyAscii As Integer)
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbDecimal, 46
        Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
        Case 45
            If Len(txtsigmam(2).Text) <> 0 Then
                KeyAscii = 0 ' ignore keystroke
                Beep
            ElseIf Len(txtsigmam(3).Text) <> 0 Then
                KeyAscii = 0
                Beep
            End If
        Case Else
            KeyAscii = 0
```

```
                Beep
        End Select
End Sub
Private Sub txtspeed_KeyPress(KeyAscii As Integer)
        Select Case KeyAscii
                Case vbKey0 To vbKey9
                Case vbDecimal, 46
                Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
                Case 45
                        If Len(txtspeed.Text) <> 0 Then
                                KeyAscii = 0 ' ignore keystroke
                                Beep
                        End If
                Case Else
                        KeyAscii = 0
                        Beep
        End Select
End Sub
Private Sub txttch_KeyPress(Index As Integer, KeyAscii As Integer)
        Select Case KeyAscii
                Case vbKey0 To vbKey9
                Case vbDecimal, 46
                Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
                Case 45
                        If Len(txttch(2).Text) <> 0 Then
                                KeyAscii = 0 ' ignore keystroke
                                Beep
                        ElseIf Len(txt tch(3).Text) <> 0 Then
                                KeyAscii = 0
                                Beep
                        End If
                Case Else
                        KeyAscii = 0
                        Beep
        End Select
End Sub
Private Sub txttcm_KeyPress(Index As Integer, KeyAscii As Integer)
```

```
        Select Case KeyAscii
            Case vbKey0 To vbKey9
            Case vbDecimal, 46
            Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
            Case 45
                If Len(txttcm(2).Text) <> 0 Then
                    KeyAscii = 0 ' ignore keystroke
                    Beep
                ElseIf Len(txttcm(3).Text) <> 0 Then
                    KeyAscii = 0
                    Beep
                End If
            Case Else
                KeyAscii = 0
                Beep
        End Select
End Sub
Private Sub txttct_KeyPress(Index As Integer, KeyAscii As Integer)
        Select Case KeyAscii
            Case vbKey0 To vbKey9
            Case vbDecimal, 46
            Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
            Case 45
                If Len(txttct(2).Text) <> 0 Then
                    KeyAscii = 0 ' ignore keystroke
                    Beep
                ElseIf Len(txttct(3).Text) <> 0 Then
                    KeyAscii = 0
                    Beep
                End If
            Case Else
                KeyAscii = 0
               Beep
        End Select
End Sub
Private Sub txtTt1_KeyPress(KeyAscii As Integer)
        Select Case KeyAscii
```

```
            Case vbKey0 To vbKey9
            Case vbDecimal, 46
            Case vbKeyBack, vbKeyClear, vbKeyTab, vbKeyUp, vbKeyDown
            Case 45
                If Len(txtTt1.Text) <> 0 Then
                    KeyAscii = 0 ' ignore keystroke
                    Beep
                End If
            Case Else
                KeyAscii = 0
                Beep
        End Select
End Sub
```

5. Blade Number Input Screen Code

```
Option Explicit
Option Base 1
Private x As Variant
Private Sub OKButton_Click()
    x = txtZ.Text
    If IsNumeric(x) = False Then
        MsgBox "Input is not a number!", vbExclamation, "Numeric Validation"
    Else
        frmZ.Hide
    End If
End Sub
```

6. About Screen Code

```
Option Explicit
' Reg Key Security Options...
Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
```

```
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE +
        _ KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS +
        _ KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL
' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1                        ' Unicode nul terminated string
Const REG_DWORD = 4                     ' 32-bit number
Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey
        As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As
        Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByVal
        hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, ByRef
        lpType As Long, ByVal lpData As String, ByRef lpcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long
Private Sub cmdSysInfo_Click()
        Call StartSysInfo
End Sub
Private Sub cmdOK_Click()
        Unload Me
End Sub
Private Sub Form_Load()
        Me.Caption = "About " & App.Title
        lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." & App.Revision
        lblTitle.Caption = App.Title
        lblDescription.Caption = App.FileDescription
End Sub

Public Sub StartSysInfo()
        On Error GoTo SysInfoErr
        Dim rc As Long
        Dim SysInfoPath As String
        ' Try To Get System Info Program Path\Name From Registry...
```

```
        If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
            gREGVALSYSINFO, SysInfoPath) Then

            ' Try To Get System Info Program Path Only From Registry...
        ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
            gREGVALSYSINFOLOC, SysInfoPath) Then

        ' Validate Existance Of Known 32 Bit File Version
            If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
                SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
                ' Error - File Can Not Be Found...
            Else
                GoTo SysInfoErr
            End If
            ' Error - Registry Entry Can Not Be Found...
        Else
            GoTo SysInfoErr
        End If
        Call Shell(SysInfoPath, vbNormalFocus)
        Exit Sub
SysInfoErr:
        MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub
Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As String,
        ByRef KeyVal As String) As Boolean
        Dim i As Long                   ' Loop Counter
        Dim rc As Long                  ' Return Code
        Dim hKey As Long                ' Handle To An Open Registry Key
        Dim hDepth As Long              '
        Dim KeyValType As Long          ' Data Type Of A Registry Key
        Dim tmpVal As String            ' Tempory Storage For A Registry Key Value
        Dim KeyValSize As Long ' Size Of Registry Key Variable
        '------------------------------------------------------------
        ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
        '------------------------------------------------------------
        rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey)
                ' Open Registry Key
        If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Handle Error...
        tmpVal = String$(1024, 0)                           ' Allocate Variable Space
        KeyValSize = 1024                                   ' Mark Variable Size
```

```
'------------------------------------------------------------
' Retrieve Registry Key Value...
'------------------------------------------------------------
rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
        KeyValType, tmpVal, KeyValSize)                 ' Get/Create Key Value
If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError         ' Handle Errors
If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then           ' Win95 Adds Null
        Terminated String...
        tmpVal = Left(tmpVal, KeyValSize - 1)           ' Null Found, Extract
                From String
Else            ' WinNT Does NOT Null Terminate String...
        tmpVal = Left(tmpVal, KeyValSize)                       ' Null Not Found,
                Extract String Only
End If
'------------------------------------------------------------
' Determine Key Value Type For Conversion...
'------------------------------------------------------------
Select Case KeyValType                  ' Search Data Types...
Case REG_SZ                             ' String Registry Key Data Type
        KeyVal = tmpVal                 ' Copy String Value
Case REG_DWORD                          ' Double Word Registry Key Data Type
        For i = Len(tmpVal) To 1 Step –1   ' Convert Each Bit
                KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))    ' Build Value Char.
                        By Char.
        Next
        KeyVal = Format$("&h" + KeyVal)         ' Convert Double Word To String
End Select
GetKeyValue = True                              ' Return Success
rc = RegCloseKey(hKey)                          ' Close Registry Key
Exit Function                                   ' Exit
GetKeyError:                    ' Cleanup After An Error Has Occured...
        KeyVal = ""             ' Set Return Val To Empty String
        GetKeyValue = False     ' Return Failure
        rc = RegCloseKey(hKey)  ' Close Registry Key
End Function
```

7. Module mdlInletCond Code

```
Option Explicit
```

```
Option Base 1
Public gamma As Double
' Specific Heat Ratio (Cp/Cv)
Public mdot As Double
' Mass Flow
Public Rbar As Double
' Specific Gas Constant
Public omega As Double
' wheel speed
Public Cp As Double
' at constant pressure
Const g As Long = 32.2
' gravitational constant
Public Capphi(3, 3) As Double
' Flow Function
Public Tt(3) As Double
' Total Temperature
Public Pt(3) As Double
' Total Pressure
Public M(3, 3) As Double
' Mach Number
Public Mw(3, 3) As Double
' Mach Relative to the Blade
Public Mz(3, 3) As Double
' Mach of the Axial Component
Public alpha(3, 3) As Double
' Inlet Flow angle
Public beta(3, 3) As Double
' Inlet Flow angle
Public x(3, 3) As Double
' Dimensionless Velocity
Public Xz(3, 3) As Double
' Dimensionless Velocity of the Axial Component
Public Xtheta(3, 3) As Double
' Dimensionaless Velocity along Theta
Public Xu(3, 3) As Double
' Dimensionless Velocity of the Wheel Speed Component
```

```
Public Xw(3, 3) As Double
' Dimensionless Velocity Relative to the Blade
Public rot(3) As Double
' Density
Public Vt(3) As Double
' Total Velocity
Public A(3) As Double
' Annulus Area
Public R(3, 3) As Double
' Radius
Public rht(3) As Double
' Hub to Tip Ratio
Public pitch(3) As Double
' Mean Line Pitch (Rm2/Rm1)
Public rst(3, 3) As Double
' Degree of Reaction
Public i As Integer
Public Const hub As Integer = 1
' Integer used when making hub calculations (hub = 1)
Public Const mn As Integer = 2
' Integer used when making mean line calculations (mn = 2)
Public Const tip As Integer = 3
' Integer used when making tip calculations (tip = 3)
Public Sub InletCond()
    ' Calculation of Inlet Conditions at the tip
    i = 1
    beta(i, tip) = Arccos(Mz(i, tip) / Mw(i, tip))
    M(i, tip) = Mz(i, tip) / Cos(DegToRad(alpha(i, tip)))
    x(i, tip) = Sqr((((gamma - 1) / 2) * M(i, tip) ^ 2) / (1 + (((gamma - 1) / 2) * M(i, tip) ^ 2)))
    Xz(i, tip) = x(i, tip) * Cos(DegToRad(alpha(i, tip)))
    Xtheta(i, tip) = Xz(i, tip) * Tan(DegToRad(alpha(i, tip)))
    Xu(i, tip) = Xtheta(i, tip) + Xz(i, tip) * Tan(DegToRad(beta(i, tip)))
    Xw(i, tip) = Xwfunc(Xz(i, tip), beta(i, tip))
    rot(i) = Pt(i) / (12 * Rbar * Tt(i))
    Cp = gg1(gamma) * Rbar
    Vt(i) = Sqr(2 * Cp * g * Tt(i)) * 12
    Capphi(i, tip) = x(i, tip) * (1 - x(i, tip) ^ 2) ^ (1 / (gamma - 1))
```

```
        A(i) = (mdot / (rot(i) * Vt(i))) * (1 / (Capphi(i, tip) * Cos(DegToRad(alpha(i, tip)))))
        R(i, tip) = (Xu(i, tip) * Vt(i)) / omega
        R(i, hub) = Sqr(R(i, tip) ^ 2 - A(i) / (22 / 7))
        rht(i) = rhtfunc(R(i, tip), R(i, hub))
        ' Mean Calculations
        R(i, mn) = rmfunc(R(i, tip), R(i, hub))
        Xtheta(i, mn) = Xthetafunc(R(i, tip), R(i, mn), Xtheta(i, tip))
        Xz(i, mn) = Xz(i, tip)
        alpha(i, mn) = alphafunc(Xtheta(i, mn), Xz(i, mn))
        Xu(i, mn) = Xufunc(R(i, tip), R(i, mn), Xu(i, tip))
        beta(i, mn) = betafunc(Xu(i, mn), Xtheta(i, mn), Xz(i, mn))
        x(i, mn) = Xfunc(Xz(i, mn), alpha(i, mn))
        M(i, mn) = Mach(gamma, x(i, mn))
        Xw(i, mn) = Xwfunc(Xz(i, mn), beta(i, mn))
        Mw(i, mn) = Mach(gamma, Xw(i, mn))
        Mz(i, mn) = Machz(M(i, mn), alpha(i, mn))
End Sub
```

8. Module mdlHubCalc Code

```
Option Explicit
Option Base 1
Public sigma(3, 3) As Double
' Solidity
Public Sub HubCalc(i As Integer)
        ' Hub Calculations
        Xz(i, hub) = Xz(i, mn) ' Radial Equilibrium
        Xtheta(i, hub) = Xthetafunc(R(i, mn), R(i, hub), Xtheta(i, mn))
        alpha(i, hub) = alphafunc(Xtheta(i, hub), Xz(i, hub))
        x(i, hub) = Xfunc(Xz(i, hub), alpha(i, hub))
        Xu(i, hub) = Xufunc(R(i, mn), R(i, hub), Xu(i, mn)) ' for constant wheel speed
        beta(i, hub) = betafunc(Xu(i, hub), Xtheta(i, hub), Xz(i, hub))
        Xw(i, hub) = Xwfunc(Xz(i, hub), beta(i, hub))
        M(i, hub) = Mach(gamma, x(i, hub))
        Mw(i, hub) = Mach(gamma, Xw(i, hub))
        Mz(i, hub) = Machz(M(i, hub), alpha(i, hub))
End Sub
```

9. Module mdlMeanCalc Code

```
Option Explicit
Option Base 1
Public Y(3, 3) As Double
' Dimensionless Local Velocity
Public Yw(3, 3) As Double
' Dimensionless Local Velocity Relative to the Blade
Public Yz(3, 3) As Double
' Dimensionless Local Velocity along the Axial Component
Public A31 As Double
' Ratio of alpha.  Unity for a repeating stage.
Public D(3, 3) As Double
' Diffusion Factor
Public tau As Double
' Total temperature ratio
Public phi21 As Double
' Change in Axial Velocity across the Rotor.
Public phi32 As Double
' Change in Axial Velocity across the Stator.
Public phi2m As Double
' Change in Axial Velocity at the Rotor exit at the Mean Line.
Public phi1m As Double
' ratio of axial velocity to rotation velocity
Public Sub MeanCalc(i As Integer)
    ' Mean Calculations for the Rotor or Stator
    If i = 2 Then
        Dim temp As Double
        beta(i, mn) = Arcsin(SinB2(DiffA(sigma(i, mn), pitch(i), D(i, mn), phi21, beta(i
            - 1, mn)), DiffB(sigma(i, mn), pitch(i)), pitch(i)))
        Xu(i, mn) = pitch(i) * Xu(i - 1, mn)
        phi1m = Xz(i - 1, mn) / Xu(i - 1, mn)
        phi2m = phi21 * phi1m * (1 / pitch(i))
        Xz(i, mn) = phi2m * Xu(i, mn)
        Xtheta(i, mn) = Xu(i, mn) - Xz(i, mn) * Tan(DegToRad(beta(i, mn)))
        alpha(i, mn) = alphafunc(Xtheta(i, mn), Xz(i, mn))
        tau = taufunc(Xu(i, mn), Xtheta(i, mn), Xu(i - 1, mn), Xtheta(i - 1, mn))
    Else
```

```
            alpha(i, mn) = A31 * alpha(i - 2, mn)
            D(i, mn) = Diffusion(phi32, alpha(i - 1, mn), alpha(i, mn), pitch(i), sigma(i,
                mn))
            Xz(i, mn) = phi32 * Xz(i - 1, mn)
            Xtheta(i, mn) = Xz(i, mn) * Tan(DegToRad(alpha(i, mn)))
            Xu(i, mn) = Xu(i - 1, mn) * pitch(i)
            beta(i, mn) = betafunc(Xu(i, mn), Xtheta(i, mn), Xz(i, mn))
        End If
        rst(i, mn) = DofReaction(Xtheta(i - 1, mn), Xu(i - 1, mn), Xtheta(i, mn), Xu(i, mn))
        x(i, mn) = Xfunc(Xz(i, mn), alpha(i, mn))
        Xw(i, mn) = Xwfunc(Xz(i, mn), beta(i, mn))
        Y(i, mn) = Yfunc(x(i, mn), tau)
        Yw(i, mn) = Yfunc(Xw(i, mn), tau)
        M(i, mn) = Mach(gamma, Y(i, mn))
        Mw(i, mn) = Mach(gamma, Yw(i, mn))
        Mz(i, mn) = Machz(M(i, mn), alpha(i, mn))
End Sub
```

10. Module mdlStgPerformance

```
Option Explicit
Option Base 1
Public omegasftc(3) As Double
' Secondary flow and tip clearance loss
Public omegap(3) As Double
' Profile Loss
Public omegas(3) As Double
' Shock Loss
Public omegat(3) As Double
' Total Loss (secondary flow + tip clearance + profile)
Public TTt1(3) As Double
' Static to Total Temperature Ratio
Public PPt1(3) As Double
' Static to Total Pressure Ratio
Public Tr1Tt1 As Double
' Total Relative Temperature Ratio
Public Pr1Pt1 As Double
' Total Relative Pressure Ratio
```

```
Public TETt1(3) As Double
Public PEPt1(3) As Double
Public PtPt1(3) As Double
' Total Pressure Ratio
Public TtTt1(3) As Double
' Total Temperature Ratio
Public AA1(3) As Double
' Area constriction ratio
Public delta(3) As Double
' Tip Gap
Public Sub RotorPerf(i As Integer, loopcount As Integer)
        Dim ShLoss As Double
        omegap(i) = 2 * sigma(i, mn) * ((Cos(DegToRad(beta(i - 1, mn))) ^ 2) /
                (Cos(DegToRad(beta(i, mn))) ^ 3)) * (0.005 + (0.16 * (D(i, mn) ^ 4)))
        If loopcount > 1 Then
                If Mw(i - 1, mn) > 1 Then
                        ShLoss = ((1 - ShockLoss(Mw(i - 1, mn), gamma)) / (1 - (1 + ((gamma
                                - 1) / 2) * Mw(i - 1, mn) ^ 2) ^ (-1 * gg1(gamma))))
                End If
                omegas(i) = ShLoss
                omegasftc(i) = SFTC(beta(i - 1, mn), beta(i, mn), sigma(i, mn), h(i), delta(i),
                        S(i))
        Else
                omegas(i) = 0
                omegasftc(i) = 0
        End If
        omegat(i) = omegap(i) + omegasftc(i) + omegas(i)
        If loopcount = 1 Then
                TTt1(i - 1) = 1 - (x(i - 1, mn) ^ 2)
                PPt1(i - 1) = TTt1(i - 1) ^ gg1(gamma)
                Tr1Tt1 = TTt1(i - 1) + (Xw(i - 1, mn) ^ 2)
                Pr1Pt1 = Tr1Tt1 ^ gg1(gamma)
                TETt1(i - 1) = Tr1Tt1 + (Xu(i, mn) ^ 2) - (Xu(i - 1, mn) ^ 2)
                PEPt1(i - 1) = TETt1(i - 1) ^ gg1(gamma)
        End If
        PEPt1(i) = PEPt1(i - 1) - (omegat(i) * (Pr1Pt1 - PPt1(i - 1)))
        TtTt1(i) = tau
        TETt1(i) = TETt1(i - 1)
```

```
        PtPt1(i) = PEPt1(i) * ((TtTt1(i) / TETt1(i)) ^ gg1(gamma)) ' Rotor Compression ratio
        TTt1(i) = TtTt1(i) - (x(i, mn) ^ 2)
        PPt1(i) = PtPt1(i) * ((TTt1(i) / tau) ^ gg1(gamma))
        Capphi(i, mn) = Y(i, mn) * ((1 - (Y(i, mn) ^ 2)) ^ g1(gamma))
        AA1(i) = ((Capphi(i - 1, tip) * Cos(DegToRad(alpha(i - 1, tip)))) / (Capphi(i, mn) *
                Cos(DegToRad(alpha(i, mn))))) * ((Sqr(TtTt1(i))) / PtPt1(i))
        A(i) = AA1(i) * A(i - 1)
        R(i, mn) = R(i - 1, mn) * pitch(i)
        h(i) = A(i) / (2 * (22 / 7) * R(i, mn))
        rht(i) = rhtfunc2(h(i), R(i, mn))
        R(i, tip) = (2 / (1 + rht(i))) * R(i, mn)
        R(i, hub) = rht(i) * R(i, tip)
End Sub


Public Sub StatorPerf(i As Integer, loopcount As Integer)
        Dim ShLoss As Double
        omegap(i) = 2 * sigma(i, mn) * ((Cos(DegToRad(alpha(i - 1, mn))) ^ 2) /
                (Cos(DegToRad(alpha(i, mn))) ^ 3)) * (0.005 + (0.16 * (D(i, mn) ^ 4)))
        If loopcount > 1 Then
                If M(i - 1, mn) > 1 Then
                        ShLoss = ((1 - ShockLoss(M(i - 1, mn), gamma)) / (1 - (1 + ((gamma -
                                1) / 2) * M(i - 1, mn) ^ 2) ^ (-1 * gg1(gamma))))
                End If
                omegas(i) = ShLoss
                omegasftc(i) = SFTC(alpha(i - 1, mn), alpha(i, mn), sigma(i, mn), h(i), delta(i),
                        S(i))
        Else
                omegas(i) = 0
                omegasftc(i) = 0
        End If
        omegat(i) = omegap(i) + omegasftc(i) + omegas(i)
        PtPt1(i) = PtPt1(i - 1) - omegat(i) * (PtPt1(i - 1) - PPt1(i - 1))
        TtTt1(i) = tau
        TTt1(i) = TtTt1(i) - (x(i, mn) ^ 2)
        PPt1(i) = ((TTt1(i) / TtTt1(i)) ^ gg1(gamma)) * PtPt1(i)
        Capphi(i, mn) = Y(i, mn) * ((1 - (Y(i, mn) ^ 2)) ^ g1(gamma))
        AA1(i) = ((Capphi(i - 2, tip) * Cos(DegToRad(alpha(i - 2, tip)))) / (Capphi(i, mn) *
                Cos(DegToRad(alpha(i, mn))))) * ((Sqr(TtTt1(i))) / PtPt1(i))
        A(i) = AA1(i) * A(i - 2)
```

```
        R(i, mn) = R(i - 1, mn) * pitch(i)
        h(i) = A(i) / (2 * (22 / 7) * R(i, mn))
        rht(i) = rhtfunc2(h(i), R(i, mn))
        R(i, tip) = (2 / (1 + rht(i))) * R(i, mn)
        R(i, hub) = rht(i) * R(i, tip)
End Sub
```

11. Module mdlHubTipCalc Code

```
Option Explicit
Option Base 1
Dim l As Integer
Public fsigma(3) As Double

' Linear function of sigma
Public Sub HubTipCalc(i As Integer)
        ' Hub & Tip Calculations for the Rotor or Stator
        For l = 1 To 3 Step 2
                Xz(i, l) = Xz(i, mn)
                Xtheta(i, l) = (R(i, mn) * Xtheta(i, mn)) / R(i, l) ' Radial Equilibrium
                Xu(i, l) = Xufunc(R(i, mn), R(i, l), Xu(i, mn)) 'for constant wheel speed
                alpha(i, l) = alphafunc(Xtheta(i, l), Xz(i, l))
                beta(i, l) = betafunc(Xu(i, l), Xtheta(i, l), Xz(i, l))
                x(i, l) = Xfunc(Xz(i, l), alpha(i, l))
                Xw(i, l) = Xwfunc(Xz(i, l), beta(i, l))
                Y(i, l) = Yfunc(x(i, l), tau)
                Yw(i, l) = Yfunc(Xw(i, l), tau)
                M(i, l) = Mach(gamma, Y(i, l))
                Mw(i, l) = Mach(gamma, Yw(i, l))
                Mz(i, l) = Machz(M(i, l), alpha(i, l))
                If l = 1 Then
                        sigma(i, l) = sigma(i, mn) * (2 - fsigma(i))
                ElseIf l = 3 Then
                        sigma(i, l) = sigma(i, mn) * fsigma(i)
                End If
                If i = 2 Then
                        D(i, l) = Diffusion(phi21, beta(i - 1, l), beta(i, l), pitch(i), sigma(i, l))
                Else
```

```
                D(i, l) = Diffusion(phi32, alpha(i - 1, l), alpha(i, l), pitch(i), sigma(i, l))
            End If
            rst(i, l) = DofReaction(Xtheta(i - 1, l), Xu(i - 1, l), Xtheta(i, l), Xu(i, l))
        Next l
End Sub
```

12. Module mdlBladeGeometry Code

```
Option Explicit
Option Base 1
Public AR(3) As Double
' Aspect Ratio
Public C(3) As Double
' Chord
Public Crev(3) As Double
' revised chord length
Public ARrev(3) As Double
' revised aspect ratio
Public Z(3) As Double
' Number of Blades calculated
Public intZ(3) As Integer
' Number of Blades chosen by the user.
Public h(3) As Double
' Blade height
Public S(3) As Double
' blade spacing
Public Sub BladeGeo(i As Integer, loopcount As Integer)
        h(i) = R(i, tip) - R(i, hub) ' blade height
        C(i) = h(i) / AR(i) ' Calculated chord length
        Z(i) = 2 * (22 / 7) * R(i, mn) * sigma(i, mn) / C(i) ' Calculated no. of blades
        If loopcount = 1 Then
            If i = 2 Then
                frmZ.lblZ.Caption = "The calculated no. of blades in the rotor (Zr) is "
                    & Z(i)
            ElseIf i = 3 Then
                frmZ.lblZ.Caption = "The calculated no. of blades in the stator (Zs) is "
                    & Z(i)
            End If
            frmZ.Show vbModal
```

```
                intZ(i) = frmZ.txtZ.Text
        End If
        ARrev(i) = h(i) / ((2 * (22 / 7) * R(i, mn) * sigma(i, mn)) / intZ(i)) ' Revised aspect ratio
        frmZ.txtZ.Text = ""
        Crev(i) = h(i) / ARrev(i) ' Revised chord length
        S(i) = Crev(i) / sigma(i, mn) ' Blade spacing
End Sub
```

13. Module mdlIncidence Code

```
Option Explicit
Option Base 1
Public i0(6, 9) As Double, slopen(7, 9) As Double, ikit(6) As Double, i0delta0(6, 9) As Double,
        slopemn(7, 9) As Double, Kit(7, 7) As Double, d0(6, 9) As Double, slopem(7, 9) As
        Double, dkit(6) As Double
' X values for 6th order curve fit polynomial.
Public i010(3, 3) As Double
' Zero-camber incidence angle
Public d010(3, 3) As Double
' Zero-camber deviation angle
Public N(3, 3) As Double
' incidence angle slope factor n
Public ikt(3, 3) As Double
' incidence angle correction factor for thickness
Public dkt(3, 3) As Double
' deviation angle correction factor for thickness
Public i0ref(3, 3) As Double
' reference incidence angle
Public d0ref(3, 3) As Double
' reference deviation angle
Public i2d As Double
' 2 dimentional incidence angle
Public icor(3, 3) As Double
' corrected 2 dimentional incidence angle
Public dref(3, 3) As Double
' deviation angle
Public camber(3, 3) As Double
' camber angle
Public i0d0(3, 3) As Double
```

```
' variation of i0 - d0
Public ktbar(3, 3) As Double
' camber angle correction factor for thickness
Public onemn(3, 3) As Double
' camber angle slope factor 1-m+n
Public dm(3, 3) As Double
' deviation angle slope factor m
Public tc(3, 3) As Double
' thickness to cord ratio.
Public k As Integer
Dim o As Integer, p As Single
Public Xvar() As Double
Public Yvar() As Double
Public Sub Incidence(i As Integer, j As Integer, slopegraph As Boolean, ksh As Single)
        ' Variation of (i0)10-(delta0)10
        ' Load constants
        ' solidity = 0.4
        i0delta0(6, 1) = -1.72244891587855E-11
        i0delta0(5, 1) = 3.05183134666209E-09
        i0delta0(4, 1) = -2.54582401992831E-07
        i0delta0(3, 1) = 8.9308266844057E-06
        i0delta0(2, 1) = -1.64583732868095E-04
        i0delta0(1, 1) = 2.36162560875073E-02
        ' solidity = 0.6
        i0delta0(6, 2) = -1.21461996934098E-10
        i0delta0(5, 2) = 2.44632274559037E-08
        i0delta0(4, 2) = -1.89083802948353E-06
        i0delta0(3, 2) = 6.51024967552871E-05
        i0delta0(2, 2) = -1.10036026944726E-03
        i0delta0(1, 2) = 4.37180120798075E-02
        ' solidity = 0.8
        i0delta0(6, 3) = -1.56214579635869E-10
        i0delta0(5, 3) = 3.09609686430234E-08
        i0delta0(4, 3) = -2.3589555206982E-06
        i0delta0(3, 3) = 7.92058315148836E-05
        i0delta0(2, 3) = -1.27482005365209E-03
        i0delta0(1, 3) = 5.69282884198401E-02
```

' solidity = 1.0

i0delta0(6, 4) = -6.93736034940097E-11

i0delta0(5, 4) = 8.79693724809005E-09

i0delta0(4, 4) = -3.60272149557694E-07

i0delta0(3, 4) = -4.42503626629787E-07

i0delta0(2, 4) = 1.21654421718631E-05

i0delta0(1, 4) = 6.52732436619772E-02

' solidity = 1.2

i0delta0(6, 5) = 1.04197027145803E-10

i0delta0(5, 5) = -2.10810745440021E-08

i0delta0(4, 5) = 1.37089527130208E-06

i0delta0(3, 5) = -3.80616685049517E-05

i0delta0(2, 5) = 8.34875831969839E-05

i0delta0(1, 5) = 8.43313899049463E-02

' solidity = 1.4

i0delta0(6, 6) = 5.17847890355591E-11

i0delta0(5, 6) = -1.11055450426056E-08

i0delta0(4, 6) = 6.3411319506379E-07

i0delta0(3, 6) = -1.51366527276764E-05

i0delta0(2, 6) = -1.39016582579643E-04

i0delta0(1, 6) = 9.70746619623242E-02

' solidity = 1.6

i0delta0(6, 7) = -2.08323213348361E-10

i0delta0(5, 7) = 3.63757529792119E-08

i0delta0(4, 7) = -2.51178491095239E-06

i0delta0(3, 7) = 7.46391658452694E-05

i0delta0(2, 7) = -1.21453189910881E-03

i0delta0(1, 7) = 0.11686046293471

' solidity = 1.8

i0delta0(6, 8) = -1.56143738685847E-10

i0delta0(5, 8) = 2.10079058766965E-08

i0delta0(4, 8) = -9.83116558606056E-07

i0delta0(3, 8) = 6.88832778905635E-06

i0delta0(2, 8) = 6.30398362773121E-05

i0delta0(1, 8) = 0.124576532031824

' solidity = 2

i0delta0(6, 9) = -3.47201982252689E-10

```
i0delta0(5, 9) = 6.02515059199005E-08
i0delta0(4, 9) = -4.22495870910922E-06
i0delta0(3, 9) = 1.38861665249124E-04
i0delta0(2, 9) = -2.50961937126704E-03
i0delta0(1, 9) = 0.157387592875239
k = 9
ReDim curve(k)
ReDim Xvar(k)
ReDim Yvar(k)
p = 0.4
For o = 1 To k
        Yvar(o) = CurveFit2(i0delta0(1, o), i0delta0(2, o), i0delta0(3, o), i0delta0(4, o),
                i0delta0(5, o), i0delta0(6, o), beta(i, j))
        Xvar(o) = p
        p = p + 0.2
Next o
QuadCoeff k
i0d0(i, j) = Interp(sigma(i, j))
If slopegraph = True Then
        ' NACA 65-(A10)-series blades as equivalent circular arc
        ' Load constants
        ' solidity = 0.4
        slopemn(7, 1) = 2.08333333236381E-11
        slopemn(6, 1) = -4.31089743405371E-09
        slopemn(5, 1) = 3.42948717823449E-07
        slopemn(4, 1) = -1.2816870624599E-05
        slopemn(3, 1) = 1.47926864929815E-04
        slopemn(2, 1) = -3.97852565922108E-03
        slopemn(1, 1) = 0.53497377639842
        ' solidity = 0.6
        slopemn(7, 2) = -3.47222221982991E-12
        slopemn(6, 2) = 7.772435880668E-10
        slopemn(5, 2) = -5.87606836521815E-08
        slopemn(4, 2) = 1.3414189909966E-06
        slopemn(3, 2) = -5.4709838117617E-05
        slopemn(2, 2) = -2.55785261049368E-03
        slopemn(1, 2) = 0.675027681002291
```

' solidity = 0.8

slopemn(7, 3) = -5.0821976835258E-21

slopemn(6, 3) = -1.28205128154327E-10

slopemn(5, 3) = 1.20192307773159E-08

slopemn(4, 3) = -6.92016318915023E-07

slopemn(3, 3) = -3.31075165860284E-05

slopemn(2, 3) = -2.12211544896945E-03

slopemn(1, 3) = 0.750014569226494

' solidity = 1.0

slopemn(7, 4) = -6.94444444304795E-12

slopemn(6, 4) = 1.20192307712444E-09

slopemn(5, 4) = -7.6655982672591E-08

slopemn(4, 4) = 1.48528553811644E-06

slopemn(3, 4) = -4.01670539815768E-05

slopemn(2, 4) = -1.64423084743248E-03

slopemn(1, 4) = 0.794953380497141

' solidity = 1.2

slopemn(7, 5) = -2.77777777637215E-11

slopemn(6, 5) = 5.48076922735063E-09

slopemn(5, 5) = -4.03579059504722E-07

slopemn(4, 5) = 1.27039626889314E-05

slopemn(3, 5) = -2.01934244898894E-04

slopemn(2, 5) = -3.78525730411639E-04

slopemn(1, 5) = 0.824973776832628

' solidity = 1.4

slopemn(7, 6) = 6.94444443457762E-12

slopemn(6, 6) = -9.1346153644617E-10

slopemn(5, 6) = 1.5758547106115E-08

slopemn(4, 6) = 1.08027389256193E-06

slopemn(3, 6) = -7.76942489579824E-05

slopemn(2, 6) = -3.57051379069162E-04

slopemn(1, 6) = 0.850023310682502

' solidity = 1.6

slopemn(7, 7) = -1.38888888844018E-11

slopemn(6, 7) = 3.07692307570384E-09

slopemn(5, 7) = -2.60683760439084E-07

slopemn(4, 7) = 9.44930068769168E-06

```
        slopemn(3, 7) = -1.87603922285007E-04
        slopemn(2, 7) = 6.39102459899732E-04
        slopemn(1, 7) = 0.870029138228418
        ' solidity = 1.8
        slopemn(7, 8) = -6.94444444643608E-12
        slopemn(6, 8) = 1.84294871811291E-09
        slopemn(5, 8) = -1.7841880317615E-07
        slopemn(4, 8) = 6.76354894046938E-06
        slopemn(3, 8) = -1.36371890420151E-04
        slopemn(2, 8) = 3.83012708653041E-04
        slopemn(1, 8) = 0.889956294448098
        ' solidity = 2
        slopemn(7, 9) = -1.04166666679601E-11
        slopemn(6, 9) = 2.3637820511721E-09
        slopemn(5, 9) = -2.13141025329211E-07
        slopemn(4, 9) = 8.38760196941735E-06
        slopemn(3, 9) = -1.82296764080547E-04
        slopemn(2, 9) = 1.1642627027868E-03
        slopemn(1, 9) = 0.900013112661114
        k = 9
        ReDim curve(k)
        ReDim Xvar(k)
        ReDim Yvar(k)
        p = 0.4
        For o = 1 To k
            Yvar(o) = CurveFit1(slopemn(1, o), slopemn(2, o), slopemn(3, o),
                slopemn(4, o), slopemn(5, o), slopemn(6, o), slopemn(7, o),
                beta(i, j))
            Xvar(o) = p
            p = p + 0.2
        Next o
        QuadCoeff k
        onemn(i, j) = Interp(sigma(i, j))
Else
        ' Circular arc mean line blades
        ' Load constants
        ' solidity = 0.4
        slopemn(7, 1) = -4.16666666439411E-11
```

```
slopemn(6, 1) = 8.14102563624798E-09
slopemn(5, 1) = -5.80929486814364E-07
slopemn(4, 1) = 1.81949300572665E-05
slopemn(3, 1) = -2.4787150325345E-04
slopemn(2, 1) = -3.33942308314938E-03
slopemn(1, 1) = 0.350002913822904
' solidity = 0.6
slopemn(7, 2) = 6.94444443965982E-12
slopemn(6, 2) = -1.4903846145501E-09
slopemn(5, 2) = 1.06303418803688E-07
slopemn(4, 2) = -3.08493589651349E-06
slopemn(3, 2) = -5.89209358992093E-06
slopemn(2, 2) = -2.75032054841517E-03
slopemn(1, 2) = 0.545032051554482
' solidity = 0.8
slopemn(7, 3) = -1.04166666603368E-11
slopemn(6, 3) = 1.56249999787653E-09
slopemn(5, 3) = -7.29166664409364E-08
slopemn(4, 3) = 1.18371202617595E-07
slopemn(3, 3) = 2.38731068833431E-05
slopemn(2, 3) = -3.17708338894818E-03
slopemn(1, 3) = 0.649981061002052
' solidity = 1.0
slopemn(7, 4) = -6.94444443796575E-12
slopemn(6, 4) = 7.21153844973621E-10
slopemn(5, 4) = -2.93803398276893E-09
slopemn(4, 4) = -2.18458625589335E-06
slopemn(3, 4) = 3.70765356052516E-05
slopemn(2, 4) = -2.16987186519191E-03
slopemn(1, 4) = 0.715017483006385
' solidity = 1.2
slopemn(7, 5) = 3.47222221644178E-12
slopemn(6, 5) = -1.32211538505372E-09
slopemn(5, 5) = 1.4369658127289E-07
slopemn(4, 5) = -6.95767774239187E-06
slopemn(3, 5) = 1.06356110350703E-04
slopemn(2, 5) = -2.01842956960263E-03
```

```
slopemn(1, 5) = 0.760024767461402
' solidity = 1.4
slopemn(7, 6) = -6.7762635780344E-21
slopemn(6, 6) = -1.2820512967221E-10
slopemn(5, 6) = 1.20192308605827E-08
slopemn(4, 6) = -8.43531474004067E-07
slopemn(3, 6) = -1.71984251977619E-05
slopemn(2, 6) = -7.05448809185327E-04
slopemn(1, 6) = 0.794938811805281
' solidity = 1.6
slopemn(7, 7) = 1.73611111042317E-11
slopemn(6, 7) = -3.34134615158702E-09
slopemn(5, 7) = 2.297008545038E-07
slopemn(4, 7) = -7.7196241337063E-06
slopemn(3, 7) = 9.70923186827122E-05
slopemn(2, 7) = -1.42612189358715E-03
slopemn(1, 7) = 0.825036422577128
' solidity = 1.8
slopemn(7, 8) = 1.73611110991495E-11
slopemn(6, 8) = -3.37339743378823E-09
slopemn(5, 8) = 2.35309829010877E-07
slopemn(4, 8) = -7.93524184317107E-06
slopemn(3, 8) = 9.22482045950801E-05
slopemn(2, 8) = -8.26442413767836E-04
slopemn(1, 8) = 0.844992716324796
' solidity = 2
slopemn(7, 9) = 3.47222221135958E-12
slopemn(6, 9) = -5.84935897461614E-10
slopemn(5, 9) = 3.55235043048019E-08
slopemn(4, 9) = -2.03634907514072E-06
slopemn(3, 9) = 3.84146771352789E-05
slopemn(2, 9) = -6.8605780506914E-04
slopemn(1, 9) = 0.859969406337427
k = 9
ReDim curve(k)
ReDim Xvar(k)
ReDim Yvar(k)
```

```
            p = 0.4
            For o = 1 To k
                        Yvar(o) = CurveFit1(slopemn(1, o), slopemn(2, o), slopemn(3, o),
                                    slopemn(4, o), slopemn(5, o), slopemn(6, o), slopemn(7, o),
                                    beta(i, j))
                        Xvar(o) = p
                        p = p + 0.2
            Next o
            QuadCoeff k
            onemn(i, j) = Interp(sigma(i, j))
End If
' Variation of thickness-correction factor Kt for camber calculation
' Load constants
' beta1 = 10
Kit(7, 1) = 816993.464355469
Kit(6, 1) = -351150.076196289
Kit(5, 1) = 61737.9966625977
Kit(4, 1) = -5315.17809592133
Kit(3, 1) = 142.41942977427
Kit(2, 1) = 13.4581061812941
Kit(1, 1) = 6.50000014506098E-02
' beta1 = 20
Kit(7, 2) = 1077614.37939453
Kit(6, 2) = -459489.065126953
Kit(5, 2) = 79133.1385789185
Kit(4, 2) = -6710.73264049728
Kit(3, 2) = 201.572413791658
Kit(2, 2) = 12.2108848959033
Kit(1, 2) = 7.74781832012336E-02
' beta1 = 30
Kit(7, 3) = 796568.628417969
Kit(6, 3) = -338565.234250488
Kit(5, 3) = 58401.6780760498
Kit(4, 3) = -4855.18434412186
Kit(3, 3) = 103.679839050816
Kit(2, 3) = 15.0572606819245
Kit(1, 3) = 6.05909105273496E-02
' beta1 = 40
```

```
Kit(7, 4) = -183823.529541016
Kit(6, 4) = 137302.036435547
Kit(5, 4) = -32831.3536986084
Kit(4, 4) = 3879.52703175537
Kit(3, 4) = -335.438802558397
Kit(2, 4) = 26.0388210932938
Kit(1, 4) = -3.53636350253408E-02
' beta1 = 50
Kit(7, 5) = -449346.408691406
Kit(6, 5) = 260840.875805664
Kit(5, 5) = -56572.084102478
Kit(4, 5) = 6297.74699841156
Kit(3, 5) = -479.565571368521
Kit(2, 5) = 30.7062532322958
Kit(1, 5) = -7.44999987640929E-02
' beta1 = 60
Kit(7, 6) = 2879901.96362305
Kit(6, 6) = -1181513.95310303
Kit(5, 6) = 193177.319282959
Kit(4, 6) = -15656.7577867749
Kit(3, 6) = 525.58354015793
Kit(2, 6) = 8.91909924834078
Kit(1, 6) = 0.126909092425712
' beta1 = 70
Kit(7, 7) = 5187908.49780273
Kit(6, 7) = -1994626.6972876
Kit(5, 7) = 298546.694419434
Kit(4, 7) = -21587.099794791
Kit(3, 7) = 600.425619817768
Kit(2, 7) = 13.8798799771508
Kit(1, 7) = 9.83181832073486E-02
k = 7
ReDim curve(k)
ReDim Xvar(k)
ReDim Yvar(k)
p = 10
For o = 1 To k
```

```
        Yvar(o) = CurveFit1(Kit(1, o), Kit(2, o), Kit(3, o), Kit(4, o), Kit(5, o), Kit(6, o),
                Kit(7, o), tc(i, j))
        Xvar(o) = p
        p = p + 10
Next o
QuadCoeff k
ktbar(i, j) = Interp(beta(i, j))
camber(i, j) = camberfunc(beta(i, j), beta((i - 1), j), ksh, ktbar(i, j), i0d0(i, j), onemn(i, j))
' Zero-camber incidence angle
' Load constants
' solidity = 0.4
i0(6, 1) = -4.04805786872846E-14
i0(5, 1) = 3.30321577382553E-10
i0(4, 1) = -1.61193689951489E-07
i0(3, 1) = 1.60266684012811E-05
i0(2, 1) = -6.12705630658184E-04
i0(1, 1) = 3.96414794354314E-02
' solidity = 0.6
i0(6, 2) = 2.77737297393952E-10
i0(5, 2) = -5.8003011805044E-08
i0(4, 2) = 4.49158409132622E-06
i0(3, 2) = -1.58973331821244E-04
i0(2, 2) = 2.41923882140327E-03
i0(1, 2) = 3.77248126474115E-02
' solidity = 0.8
i0(6, 3) = -2.77737297421058E-10
i0(5, 3) = 5.38363451353663E-08
i0(4, 3) = -3.86658408690899E-06
i0(3, 3) = 1.2355666447661E-04
i0(2, 3) = -1.73173878386024E-03
i0(1, 3) = 7.33585199368463E-02
' solidity = 1.0
i0(6, 4) = -6.24048706766443E-10
i0(5, 4) = 1.33904109689276E-07
i0(4, 4) = -1.07952815886492E-05
i0(3, 4) = 3.96289954153417E-04
i0(2, 4) = -6.38683407169083E-03
```

```
i0(1, 4) = 0.11471689445807
' solidity = 1.2
i0(6, 5) = 1.3907105163107E-10
i0(5, 5) = -2.85697804719431E-08
i0(4, 5) = 2.01009383982154E-06
i0(3, 5) = -5.64950093888683E-05
i0(2, 5) = 3.77314309162102E-04
i0(1, 5) = 0.100321674263796
' solidity = 1.4
i0(6, 6) = -1.04116066001981E-10
i0(5, 6) = 2.04204313875023E-08
i0(4, 6) = -1.67350788249365E-06
i0(3, 6) = 6.85083302869316E-05
i0(2, 6) = -1.42057624429981E-03
i0(1, 6) = 0.123885649683871
' solidity = 1.6
i0(6, 7) = -6.59874024969446E-10
i0(5, 7) = 1.26238706023296E-07
i0(4, 7) = -9.07669856742288E-06
i0(3, 7) = 3.00725006795233E-04
i0(2, 7) = -4.56882667162972E-03
i0(1, 7) = 0.153343047833914
' solidity = 1.8
i0(6, 8) = -2.42843032737926E-10
i0(5, 8) = 3.99324784211563E-08
i0(4, 8) = -2.4176220057015E-06
i0(3, 8) = 6.48183223432852E-05
i0(2, 8) = -8.29475889986497E-04
i0(1, 8) = 0.149069730896372
' solidity = 2
i0(6, 9) = -6.5928705639891E-10
i0(5, 9) = 1.27699043138418E-07
i0(4, 9) = -9.55189003892798E-06
i0(3, 9) = 3.42296645044371E-04
i0(2, 9) = -5.90334486059874E-03
i0(1, 9) = 0.198583259410952
k = 9
```

```
ReDim curve(k)
ReDim Xvar(k)
ReDim Yvar(k)
p = 0.4
For o = 1 To k
        Yvar(o) = CurveFit2(i0(1, o), i0(2, o), i0(3, o), i0(4, o), i0(5, o), i0(6, o), beta(i,
                j))
        Xvar(o) = p
        p = p + 0.2
Next o
QuadCoeff k
i010(i, j) = Interp(sigma(i, j))
' Minimum loss incidence angle slope factor
' Load constants
' solidity = 0.4
slopen(7, 1) = 5.20833333567409E-12
slopen(6, 1) = -1.00560897412991E-09
slopen(5, 1) = 7.05128204714356E-08
slopen(4, 1) = -2.35449445540326E-06
slopen(3, 1) = 7.51638881979488E-06
slopen(2, 1) = -3.64078518850874E-03
slopen(1, 1) = -5.00123838208282E-02
' solidity = 0.6
slopen(7, 2) = -1.73611110737386E-12
slopen(6, 2) = 2.60416666312754E-10
slopen(5, 2) = -2.25694444522251E-08
slopen(4, 2) = 1.0866477326843E-06
slopen(3, 2) = -5.59643317217251E-05
slopen(2, 2) = -2.28645827399987E-03
slopen(1, 2) = -4.49905306238207E-02
' solidity = 0.8
slopen(7, 3) = 6.94444444474202E-12
slopen(6, 3) = -1.85897435877984E-09
slopen(5, 3) = 1.70806623789321E-07
slopen(4, 3) = -7.15544871177087E-06
slopen(3, 3) = 1.05446046163138E-04
slopen(2, 3) = -2.7290063578107E-03
```

slopen(1, 3) = -4.00160259214886E-02

' solidity = 1.0

slopen(7, 4) = -5.2083333280508E-12

slopen(6, 4) = 9.57532051044929E-10

slopen(5, 4) = -7.77243589489274E-08

slopen(4, 4) = 3.08220425893069E-06

slopen(3, 4) = -8.76518800509984E-05

slopen(2, 4) = -8.65945466387075E-04

slopen(1, 4) = -3.49963580541726E-02

' solidity = 1.2

slopen(7, 5) = 2.60416666588887E-11

slopen(6, 5) = -5.73317307430571E-09

slopen(5, 5) = 4.60336538235517E-07

slopen(4, 5) = -1.68285620523179E-05

slopen(3, 5) = 2.38823935717392E-04

slopen(2, 5) = -2.12556085602955E-03

slopen(1, 5) = -2.99817892565741E-02

' solidity = 1.4

slopen(7, 6) = 1.56249999943168E-11

slopen(6, 6) = -3.43349358927075E-09

slopen(5, 6) = 2.74038461367532E-07

slopen(4, 6) = -1.01127258078648E-05

slopen(3, 6) = 1.33344623499454E-04

slopen(2, 6) = -1.06402239998715E-03

slopen(1, 6) = -2.49992717449814E-02

' solidity = 1.6

slopen(7, 7) = 1.56249999951638E-11

slopen(6, 7) = -3.32131410091599E-09

slopen(5, 7) = 2.54407051133998E-07

slopen(4, 7) = -8.82776077215652E-06

slopen(3, 7) = 9.46172051285998E-05

slopen(2, 7) = -2.87900609066583E-04

slopen(1, 7) = -1.99978148450413E-02

' solidity = 1.8

slopen(7, 8) = -5.20833332889783E-12

slopen(6, 8) = 1.11778846096679E-09

slopen(5, 8) = -9.01442307604805E-08

```
slopen(4, 8) = 2.80612616787579E-06
slopen(3, 8) = -5.87438087080727E-05
slopen(2, 8) = 3.75240412154199E-04
slopen(1, 8) = -0.014986159811361
' solidity = 2
slopen(7, 9) = 1.21527777732163E-11
slopen(6, 9) = -2.31971153754913E-09
slopen(5, 9) = 1.6159188022391E-07
slopen(4, 9) = -5.68345716178698E-06
slopen(3, 9) = 7.84089690597511E-05
slopen(2, 9) = -3.66426257727426E-04
slopen(1, 9) = -9.98615979092321E-03
k = 9
ReDim curve(k)
ReDim Xvar(k)
ReDim Yvar(k)
p = 0.4
For o = 1 To k
    Yvar(o) = CurveFit1(slopen(1, o), slopen(2, o), slopen(3, o), slopen(4, o),
        slopen(5, o), slopen(6, o), slopen(7, o), beta(i, j))
    Xvar(o) = p
    p = p + 0.2
Next o
QuadCoeff k
N(i, j) = Interp(sigma(i, j))
' Maximum thickness correction factor
' Load constants
ikit(6) = -748795.365783691
ikit(5) = 243951.67288208
ikit(4) = -28087.2979736328
ikit(3) = 1612.26135325431
ikit(2) = -137.429116554558
ikit(1) = 18.8187430176185
ikt(i, j) = CurveFit2(ikit(1), ikit(2), ikit(3), ikit(4), ikit(5), ikit(6), tc(i, j))
i0ref(i, j) = ksh * ikt(i, j) * i010(i, j)
i2d = i0ref(i, j) + N(i, j) * camber(i, j)
If i = 2 Then
```

```
        If ksh = 0.7 Then
                icor(i, j) = 0.7238 * Mw(i - 1, j) + 7.5481 + i2d
        ElseIf ksh = 1.1 Then
                icor(i, j) = 1.3026 * Mw(i - 1, j) + 5.738 + i2d
        ElseIf ksh = 1 Then
                icor(i, j) = i2d
        End If
ElseIf i = 3 Then
        If ksh = 0.7 Then
                icor(i, j) = 0.7238 * M(i - 1, j) + 7.5481 + i2d
        ElseIf ksh = 1.1 Then
                icor(i, j) = 1.3026 * M(i - 1, j) + 5.738 + i2d
        ElseIf ksh = 1 Then
                icor(i, j) = i2d
        End If
End If
' Zero-camber deviation angle
' Load constants
' solidity = 0.4
d0(6, 1) = -6.95861265439764E-11
d0(5, 1) = 1.57394588647108E-08
d0(4, 1) = -1.36278903151155E-06
d0(3, 1) = 5.7135006738207E-05
d0(2, 1) = -1.06391418466956E-03
d0(1, 1) = 1.59951786608872E-02
' solidity = 0.6
d0(6, 2) = -7.08410313374729E-14
d0(5, 2) = -4.63603904397869E-10
d0(4, 2) = 1.86661040402214E-07
d0(3, 2) = -1.62241633923088E-05
d0(2, 2) = 6.30890134971196E-04
d0(1, 2) = 5.06008922957335E-03
' solidity = 0.8
d0(6, 3) = -1.72852100539214E-11
d0(5, 3) = 3.54731371186856E-09
d0(4, 3) = -1.83872938541718E-07
d0(3, 3) = 1.7208295339799E-06
```

d0(2, 3) = 2.60107809708643E-04

d0(1, 3) = 9.95347546358971E-03

' solidity = 1.0

d0(6, 4) = 1.91189805496704E-10

d0(5, 4) = -3.92754785100841E-08

d0(4, 4) = 3.08863831766093E-06

d0(3, 4) = -1.10622510739233E-04

d0(2, 4) = 2.00666089165225E-03

d0(1, 4) = 9.99963312096952E-04

' solidity = 1.2

d0(6, 5) = -2.95336231991993E-10

d0(5, 5) = 6.09853177350322E-08

d0(4, 5) = -4.62262480382947E-06

d0(3, 5) = 1.62505009349445E-04

d0(2, 5) = -2.28780804479811E-03

d0(1, 5) = 2.54709629980425E-02

' solidity = 1.4

d0(6, 6) = -1.75483338526582E-11

d0(5, 6) = 1.52773729431011E-09

d0(4, 6) = 2.26701400096729E-07

d0(3, 6) = -2.11891579624535E-05

d0(2, 6) = 8.81687814285215E-04

d0(1, 6) = 7.87309303404982E-03

' solidity = 1.6

d0(6, 7) = -3.46716215172916E-11

d0(5, 7) = 7.92043135938725E-09

d0(4, 7) = -5.62396773595708E-07

d0(3, 7) = 2.2674997246952E-05

d0(2, 7) = -1.57381818780777E-04

d0(1, 7) = 0.017343983408864

' solidity = 1.8

d0(6, 8) = 6.30485119950015E-11

d0(5, 8) = -1.41425240546278E-08

d0(4, 8) = 1.37167452307629E-06

d0(3, 8) = -5.62863600421792E-05

d0(2, 8) = 1.32778646161569E-03

d0(1, 8) = 8.46711704980407E-03

```
' solidity = 2
d0(6, 9) = 8.67296545047297E-11
d0(5, 9) = -1.60473137209016E-08
d0(4, 9) = 1.29498405332384E-06
d0(3, 9) = -4.33874967669112E-05
d0(2, 9) = 8.78086666489253E-04
d0(1, 9) = 1.43381905636488E-02
k = 9
ReDim curve(k)
ReDim Xvar(k)
ReDim Yvar(k)
p = 0.4
For o = 1 To k
    Yvar(o) = CurveFit2(d0(1, o), d0(2, o), d0(3, o), d0(4, o), d0(5, o), d0(6, o),
        beta(i, j))
    Xvar(o) = p
    p = p + 0.2
Next o
QuadCoeff k
d010(i, j) = Interp(sigma(i, j))
If slopegraph = True Then
    ' NACA 65-(A10)-series blades as equivalent circular arc
    ' Load constants
    ' solidity = 0.4
    slopem(7, 1) = 6.94444443457762E-12
    slopem(6, 1) = -1.49038461433326E-09
    slopem(5, 1) = 1.06303418845322E-07
    slopem(4, 1) = -3.16069348027526E-06
    slopem(3, 1) = 7.70624526893471E-05
    slopem(2, 1) = -7.91987265529315E-04
    slopem(1, 1) = 0.412494173012831
    ' solidity = 0.6
    slopem(7, 2) = 2.08333333206735E-11
    slopem(6, 2) = -4.48717948388355E-09
    slopem(5, 2) = 3.52964743544071E-07
    slopem(4, 2) = -1.22224650365155E-05
    slopem(3, 2) = 2.05185752662373E-04
```

```
slopem(2, 2) = -8.88621860667627E-04
slopem(1, 2) = 0.277498543514405
' solidity = 0.8
slopem(7, 3) = 1.04166666611838E-11
slopem(6, 3) = -2.20352563908184E-09
slopem(5, 3) = 1.64262820512295E-07
slopem(4, 3) = -4.77163461454211E-06
slopem(3, 3) = 6.08733981977139E-05
slopem( 2, 3) = 2.12339688573593E-04
slopem(1, 3) = 0.210016025961522
' solidity = 1.0
slopem(7, 4) = 3.47222221813584E-12
slopem(6, 4) = -1.19391025581519E-09
slopem(5, 4) = 1.31677350481696E-07
slopem(4, 4) = -5.84899476141487E-06
slopem(3, 4) = 1.26963627380405E-04
slopem(2, 4) = -5.62980817107928E-04
slopem(1, 4) = 0.170010198410182
' solidity = 1.2
slopem(7, 5) = 5.2083333305919E-12
slopem(6, 5) = -1.51842948674721E-09
slopem(5, 5) = 1.55048076988518E-07
slopem(4, 5) = -6.59036276573488E-06
slopem(3, 5) = 1.3337230542021E-04
slopem(2, 5) = -5.56330170923047E-04
slopem(1, 5) = 0.142489073668344
' solidity = 1.4
slopem(7, 6) = 2.25694444314355E-11
slopem(6, 6) = -4.79567307436549E-09
slopem(5, 6) = 3.83947649437721E-07
slopem(4, 6) = -1.38976908479194E-05
slopem( 3, 6) = 2.37974820819886E-04
slopem(2, 6) = -1.0776442692304E-03
slopem(1, 6) = 0.122499271780747
' solidity = 1.6
slopem(7, 7) = -1.04166666620308E-11
slopem(6, 7) = 1.94711538320687E-09
```

```
slopem(5, 7) = -1.29807692192396E-07
slopem(4, 7) = 4.01260197557818E-06
slopem(3, 7) = -4.06300984678865E-05
slopem(2, 7) = 3.55929451188786E-04
slopem(1, 7) = 0.107513112087279
' solidity = 1.8
slopem(7, 8) = 1.21527777736398E-11
slopem(6, 8) = -2.80048076742312E-09
slopem(5, 8) = 2.45726495706755E-07
slopem(4, 8) = -9.71317744902223E-06
slopem(3, 8) = 1.83019983943211E-04
slopem(2, 8) = -8.29567340645099E-04
slopem(1, 8) = 9.50021854990979E-02
' solidity = 2
slopem(7, 9) = -3.47222222152398E-12
slopem(6, 9) = 6.00961538020119E-10
slopem(5, 9) = -3.31196580577453E-08
slopem(4, 9) = 8.84688224900287E-07
slopem(3, 9) = 4.18366859378239E-07
slopem(2, 9) = 1.88301250858558E-04
slopem(1, 9) = 8.74956295422606E-02
k = 9
ReDim curve(k)
ReDim Xvar(k)
ReDim Yvar(k)
p = 0.4
For o = 1 To k
        Yvar(o) = CurveFit1(slopem(1, o), slopem(2, o), slopem(3, o),
                slopem(4, o), slopem(5, o), slopem(6, o), slopem(7, o), beta(i,
                j))
        Xvar(o) = p
        p = p + 0.2
Next o
QuadCoeff k
dm(i, j) = Interp(sigma(i, j))
Else
' Circular arc mean line blades
' Load constants
```

```
' solidity = 0.4
slopem(7, 1) = -5.0821976835258E-21
slopem(6, 1) = 6.73076922756005E-10
slopem(5, 1) = -1.1258012813653E-07
slopem(4, 1) = 6.10941141054866E-06
slopem(3, 1) = -1.28908069711997E-04
slopem(2, 1) = 1.04214733619301E-03
slopem(1, 1) = 0.602489802490481
' solidity = 0.6
slopem(7, 2) = 5.20833332212157E-12
slopem(6, 2) = -1.08573717811505E-09
slopem(5, 2) = 8.45352564893265E-08
slopem(4, 2) = -3.08293269490889E-06
slopem(3, 2) = 5.90424689903557E-05
slopem(2, 2) = -9.9919948297611E-05
slopem(1, 2) = 0.409991987649079
' solidity = 0.8
slopem(7, 3) = 1.73611110229166E-12
slopem(6, 3) = -1.9631410169349E-10
slopem(5, 3) = 9.34829047505303E-10
slopem(4, 3) = 6.51405884788403E-07
slopem(3, 3) = -1.73966821961358E-05
slopem(2, 3) = 6.07932628156505E-04
slopem(1, 3) = 0.310002185699076
' solidity = 1.0
slopem(7, 4) = -6.94444444474202E-12
slopem(6, 4) = 1.66666666600625E-09
slopem(5, 4) = -1.47569444275408E-07
slopem(4, 4) = 6.10795453948043E-06
slopem(3, 4) = -1.06196337540609E-04
slopem(2, 4) = 1.0979166113998E-03
slopem(1, 4) = 0.249981060934445
' solidity = 1.2
slopem(7, 5) = -1.21527777719458E-11
slopem(6, 5) = 2.57612179309884E-09
slopem(5, 5) = -2.01255341752971E-07
slopem(4, 5) = 7.28529282589818E-06
```

```
slopem(3, 5) = -1.10063251923975E-04
slopem(2, 5) = 1.03774033499349E-03
slopem(1, 5) = 0.210013840616803
' solidity = 1.4
slopem(7, 6) = -6.94444444135388E-12
slopem(6, 6) = 1.53846153871928E-09
slopem(5, 6) = -1.19925213509786E-07
slopem(4, 6) = 4.09965034364745E-06
slopem(3, 6) = -4.48436278688291E-05
slopem(2, 6) = 3.82051237181713E-04
slopem(1, 6) = 0.182514569024661
' solidity = 1.6
slopem(7, 7) = -2.25694444348236E-11
slopem(6, 7) = 4.50721153650614E-09
slopem(5, 7) = -3.28258546761218E-07
slopem(4, 7) = 1.07734192780207E-05
slopem(3, 7) = -1.40615408163569E-04
slopem(2, 7) = 8.60176240280452E-04
slopem(1, 7) = 0.160005099306115
' solidity = 1.8
slopem(7, 8) = 1.73611110991495E-12
slopem(6, 8) = -5.48878204605779E-10
slopem(5, 8) = 6.78418804189729E-08
slopem(4, 8) = -3.58591929172647E-06
slopem(3, 8) = 9.18654094732574E-05
slopem(2, 8) = -3.68509653696947E-04
slopem(1, 8) = 0.142494901151593
' solidity = 2
slopem(7, 9) = -5.20833333567409E-12
slopem(6, 9) = 1.14983974338484E-09
slopem(5, 9) = -9.05448717253288E-08
slopem(4, 9) = 3.42894084548462E-06
slopem(3, 9) = -5.49887086620515E-05
slopem(2, 9) = 7.85977528323656E-04
slopem(1, 9) = 0.1275007286422
k = 9
ReDim curve(k)
```

```
                ReDim Xvar(k)
                ReDim Yvar(k)
                p = 0.4
                For o = 1 To k
                        Yvar(o) = CurveFit1(slopem(1, o), slopem(2, o), slopem(3, o),
                                slopem(4, o), slopem(5, o), slopem(6, o), slopem(7, o), beta(i,
                                j))
                        Xvar(o) = p
                        p = p + 0.2
                Next o
                QuadCoeff k
                dm(i, j) = Interp(sigma(i, j))
        End If
        ' Maximum thickness correction factor
        ' Load constants
        dkit(6) = 618823.625244141
        dkit(5) = -202775.302703857
        dkit(4) = 25013.8597869873
        dkit(3) = -1269.01561832427
        dkit(2) = 41.3428950682282
        dkit(1) = 7.56794627627824
        dkt(i, j) = CurveFit2(ikit(1), ikit(2), ikit(3), ikit(4), ikit(5), ikit(6), tc(i, j))
        d0ref(i, j) = ksh * dkt(i, j) * d010(i, j)
        dref(i, j) = d0ref(i, j) + dm(i, j) * camber(i, j)
End Sub
```

14. Module mdlInterpolation Code

```
Option Explicit
Option Base 1
Public Ainterp() As Double
Public Binterp() As Double
Public Cinterp() As Double
Dim o As Integer
Public Sub QuadCoeff(k As Integer)
        ReDim Ainterp(k)
        ReDim Binterp(k)
        ReDim Cinterp(k)
        For o = 2 To k - 1
```

```
            Ainterp(o) = interpA(Xvar(o + 1), Xvar(o - 1), Xvar(o), Yvar(o), Yvar(o + 1),
                Yvar(o - 1))
            Binterp(o) = interpB(Yvar(o), Yvar(o - 1), Xvar(o), Xvar(o - 1), Ainterp(o))
            Cinterp(o) = interpC(Yvar(o), Ainterp(o), Binterp(o), Xvar(o))
        Next o
        Ainterp(1) = Ainterp(2)
        Binterp(1) = Binterp(2)
        Cinterp(1) = Cinterp(2)
        Ainterp(k) = Ainterp(k - 1)
        Binterp(k) = Binterp(k - 1)
        Cinterp(k) = Cinterp(k - 1)
        For o = 1 To k - 1
            Ainterp(o) = (Ainterp(o) + Ainterp(o + 1)) / 2
            Binterp(o) = (Binterp(o) + Binterp(o + 1)) / 2
            Cinterp(o) = (Cinterp(o) + Cinterp(o + 1)) / 2
        Next o
End Sub
Public Function Interp(Xval As Double)
        o = 2
        Do
            If Xval < Xvar(o) Then
                Interp = Ainterp(o) * Xval ^ 2 + Binterp(o) * Xval + Cinterp(o)
                Exit Do
            Else
                If Xval > Xvar(o) And Xval < Xvar(o + 1) Then
                    Interp = Ainterp(o) * Xval ^ 2 + Binterp(o) * Xval +
                        Cinterp(o)
                    Exit Do
                Else
                    o = o + 1
                End If
            End If
            If o = k - 1 Then
                Interp = Ainterp(o) * Xval ^ 2 + Binterp(o) * Xval + Cinterp(o)
                Exit Do
            End If
        Loop Until o = k
End Function
```

15. Module mdlFunctions Code

```
Option Exp licit

Option Base 1

Public Function Diffusion(phi1 As Double, beta1 As Double, beta2 As Double, pitch_ml As
        Double, solidity As Double) As Double

        Diffusion = 1 - phi1 * (Cos(DegToRad(beta1)) / Cos(DegToRad(beta2))) +
                (((Tan(DegToRad(beta1)) - pitch_ml * phi1 * Tan(DegToRad(beta2))) *
                Cos(DegToRad(beta1))) / ((1 + pitch_ml) * solidity))

End Function

Public Function gg1(g As Double) As Double

        gg1 = g / (g - 1)

End Function

Public Function g1(g As Double) As Double

        g1 = 1 / (g - 1)

End Function

Public Function g1g(g As Double) As Double

        g1g = (g - 1) / g

End Function

Public Function Mach(g As Double, X1 As Double) As Double

        Mach = Sqr(((2 / (g - 1)) * (X1 ^ 2)) / (1 - (X1 ^ 2)))

End Function

Public Function DegToRad(angle As Double) As Double

        DegToRad = angle * ((22 / 7) / 180)

End Function

Public Function RadToDeg(radians As Double) As Double

   RadToDeg = radians * (180 / (22 / 7))

End Function

Public Function Arccos(ratio As Double) As Double

   Arccos = RadToDeg(Atn(-ratio / Sqr(-ratio * ratio + 1)) + 2 * Atn(1))

End Function

Public Function Arcsin(ratio As Double) As Double

   Arcsin = RadToDeg(Atn(ratio / Sqr(-ratio * ratio + 1)))

End Function

Public Function DofReaction(Xtheta1 As Double, Xu1 As Double, Xtheta2 As Double, XU2 As
Double) As Double

   DofReaction = (1 - 0.5 * ((Xtheta1 / Xu1) + (Xtheta2 / XU2)))

End Function
```

```
Public Function SFTC(beta1 As Double, beta2 As Double, solidity As Double, height As Double,
        tipgap As Double, spacing As Double)

        Dim CDi As Double

        Dim CL As Double

        Dim betainf As Double

        betainf = Atn((Tan(DegToRad(beta1)) + Tan(DegToRad(beta2))) / 2)

        CL = (2 / solidity) * (Tan(DegToRad(beta1)) - Tan(DegToRad(beta2))) *
                Cos(DegToRad(betainf))

        CDi = (0.25 * (CL ^ 2) * solidity * (tipgap / height) * (1 / Cos(DegToRad(beta2)))) +
                (0.04 * (CL ^ 2) * solidity * (spacing / height))

        SFTC = CDi * ((Cos(DegToRad(beta1)) ^ 2) / (Cos(DegToRad(betainf)) ^ 3)) * solidity

End Function

Public Function taufunc(XU2 As Double, Xtheta2 As Double, Xu1 As Double, Xtheta1 As
        Double) As Double

        taufunc = 1 + 2 * ((XU2 * Xtheta2) - (Xu1 * Xtheta1))

End Function

Public Function Xfunc(Xz1 As Double, alpha1 As Double) As Double

        Xfunc = Xz1 / Cos(DegToRad(alpha1))

End Function

Public Function Xthetafunc(r1 As Double, r2 As Double, Xtheta1 As Double) As Double

        Xthetafunc = (r1 / r2) * Xtheta1

End Function

Public Function Xufunc(r1 As Double, r2 As Double, Xu1 As Double) As Double

        Xufunc = (r2 / r1) * Xu1

End Function

Public Function Xwfunc(Xz1 As Double, beta1 As Double) As Double

        Xwfunc = Xz1 / Cos(DegToRad(beta1))

End Function

Public Function Yfunc(X1 As Double, tau1 As Double) As Double

        Yfunc = X1 / Sqr(tau1)

End Function

Public Function Machz(M1 As Double, alpha1 As Double) As Double

        Machz = M1 * Cos(DegToRad(alpha1))

End Function

Public Function alphafunc(Xtheta1 As Double, Xz1 As Double) As Double

        alphafunc = RadToDeg(Atn(Xtheta1 / Xz1))

End Function

Public Function betafunc(Xu1 As Double, Xtheta1 As Double, Xz1 As Double) As Double
```

```
    betafunc = RadToDeg(Atn((Xu1 - Xtheta1) / Xz1))
End Function
Public Function rhfunc1(rt As Double, A1 As Double) As Double
    rhfunc1 = Sqr(rt ^ 2 - (A1 / (22 / 7)))
End Function
Public Function rhfunc2(rm As Double, rt As Double) As Double
    rhfunc2 = 2 * rm - rt
End Function
Public Function rtfunc(A1 As Double, rm As Double) As Double
    rtfunc = (A1 / (4 * rm * (22 / 7))) + rm
End Function
Public Function rmfunc(rt As Double, rh As Double) As Double
    rmfunc = (rt + rh) / 2
End Function
Public Function rhtfunc(rt As Double, rh As Double) As Double
    rhtfunc = rh / rt
End Function
Public Function rhtfunc2(h As Double, rm As Double) As Double
    Dim hrm As Double
    hrm = h / (2 * rm)
    rhtfunc2 = (1 - hrm) / (1 + hrm)
End Function
Public Function CurveFit1(const0 As Double, const1 As Double, const2 As Double, const3 As
    Double, const4 As Double, const5 As Double, const6 As Double, polyvar As Double)
    ' No y intercept
    CurveFit1 = (const6 * (polyvar ^ 6)) + (const5 * (polyvar ^ 5)) + (const4 * (polyvar ^ 4))
        + (const3 * (polyvar ^ 3)) + (const2 * (polyvar ^ 2)) + (const1 * polyvar) +
        const0
End Function
Public Function CurveFit2(const1 As Double, const2 As Double, const3 As Double, const4 As
    Double, const5 As Double, const6 As Double, polyvar As Double)
    ' y intercept
    CurveFit2 = (const6 * (polyvar ^ 6)) + (const5 * (polyvar ^ 5)) + (const4 * (polyvar ^ 4))
        + (const3 * (polyvar ^ 3)) + (const2 * (polyvar ^ 2)) + (const1 * polyvar)
End Function
Public Function camberfunc(beta2 As Double, beta1 As Double, kshape As Single, kthick As
    Double, idelta As Double, slope As Double)
    camberfunc = ((beta2 - beta1) - (kshape * kthick * idelta)) / slope
End Function
```

```
Public Function interpA(Xip1 As Double, Xim1 As Double, Xi As Double, Yi As Double, Yip1
        As Double, Yim1 As Double)

        interpA = (1 / (Xip1 - Xim1)) * (((Yip1 - Yi) / (Xip1 - Xi)) - ((Yi - Yim1) / (Xi - Xim1)))

End Function

Public Function interpB(Yi As Double, Yim1 As Double, Xi As Double, Xim1 As Double, Ai As
        Double)

        interpB = ((Yi - Yim1) / (Xi - Xim1)) - Ai * (Xi + Xim1)

End Function

Public Function interpC(Yi As Double, Ai As Double, Bi As Double, Xi As Double)

        interpC = Yi - Ai * Xi ^ 2 - Bi * Xi

End Function

Public Function ShockLoss(M As Double, g As Double)

        Dim ystar As Double

        ystar = (1 / (4 * g * M ^ 2)) * (((g + 1) * M ^ 2) - (3 - g) + Sqr((g + 1) * (((g + 1) * M ^ 4)
                - (2 * (3 - g) * M ^ 2) + g + 9)))

        ShockLoss = ((g + 1) / (2 * g * M ^ 2 * ystar - (g - 1))) ^ g1(g) * (((g + 1) * M ^ 2 *
                ystar) / (2 + (g - 1) * M ^ 2 * ystar)) ^ gg1(g)

End Function

Public Function DiffB(sigma As Double, R As Double)

        DiffB = 2 * sigma * R * (1 + R)

End Function

Public Function DiffA(sigma As Double, R As Double, D As Double, phi As Double, beta1 As
        Double)

        DiffA = (1 + R) * sigma * ((1 - D) / (phi * Cos(DegToRad(beta1)))) +
                (Tan(DegToRad(beta1)) / phi)

End Function

Public Function SinB2(A As Double, B As Double, R As Double)

        SinB2 = (-B + Sqr(B ^ 2 + (4 * (A ^ 2 - (B ^ 2 / (4 * R ^ 2))) * (R ^ 2 + A ^ 2)))) / (2 * (R
                ^ 2 + A ^ 2))

End Function
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. CALCULATION RESULTS

C1. HAND CALCULATIONS

Inputs derived from Sanger [Ref. 13].

$\dot{m} = 17.09 \text{lbm/sec}$

$P_{t1} = 14.69 \text{psia}$

$T_{t1} = 517.8^\circ \text{R}$

$M_{W1t} = 1.28$

$\omega = 2836.33 \text{rad/sec}$

$R = 53.35 \text{ft} \cdot \text{lbsf / lbm} \cdot {}^\circ \text{R}$

$\gamma = 1.4$

$\alpha_{1t} = 0^\circ$

$U_t = \omega \cdot r_t = 2836.33 \cdot (5.5 \text{ inches/12 inches per ft}) = 1300 \text{ ft/sec}$

$$X_{U1t} = \frac{U_{1t}}{\sqrt{2 \cdot C_P \cdot g \cdot T_{t1}}} \text{ where } C_P = \left(\frac{\gamma}{\gamma - 1}\right) \cdot R = \left(\frac{1.4}{1.4 - 1}\right) \cdot 53.35 = 186.725$$

$$\therefore \quad X_{U1t} = \frac{1300}{\sqrt{2 \cdot 186.725 \cdot 32.2 \cdot 517.8}} = 0.5210$$

$$\sin^2 \beta_{1t} = \left(\frac{X_{U1t}^2}{1 + X_{U1t}^2}\right) \cdot \left(\frac{1 + \frac{\gamma - 1}{2} \cdot M_{W1t}^2}{\frac{\gamma - 1}{2} \cdot M_{W1t}^2}\right) = \left(\frac{0.5210^2}{1 + 0.5210^2}\right) \cdot \left(\frac{1 + \frac{1.4 - 1}{2} \cdot 1.28^2}{\frac{1.4 - 1}{2} \cdot 1.28^2}\right)$$

$\sin^2 \beta_{1t} = 0.8649$

$\sin \beta_{1t} = 0.9300 \Rightarrow \beta_{1t} = \sin^{-1}(0.9300) = 68.438^\circ$

$M_{Z1t} = M_{W1t} \cdot \cos \beta_{1t} = 1.28 \cdot \cos(68.438^\circ) = 0.4704$

$D_{2m} = 0.455$; $R_{21} = 1.0687$; $\phi_{21} = 0.956$

$\sigma_{2m} = 1.52$; $f_{\sigma_2}(\sigma_{2t} / \sigma_{2m}) = 0.8553$

$\delta_2 = 0.0045$; $AR_2 = 1.2$

$(t/c)_{2max} = 0.037$ at tip; 0.056 at mean; 0.08 at hub

$D_{3m} = 0.52$; $R_{32} = 1.0259$; $\phi_{32} = 1.0371$

$\sigma_{3m} = 1.25$; $f_{\sigma_3}(\sigma_{3t} / \sigma_{3m}) = 0.8$

$\delta_3 = 0.003$; $AR_3 = 1.2$

$(t/c)_{3max} = 0.07$ at tip; 0.06 at mean; 0.05 at hub

Applying equations from Appendix A yields the followings results.

1. Inlet Conditions

$$M_{1t} = \frac{0.4704}{\cos 0} = 0.4704$$

$$X_{1t} = \sqrt{\frac{\frac{1.4-1}{2}\cdot 0.4704^2}{1+\frac{1.4-1}{2}\cdot 0.4704^2}} = 0.2059$$

$$X_{Z1t} = (0.2059)\cdot \cos 0 = 0.2059$$

$$X_{\theta 1t} = (0.2059)\cdot \tan 0 = 0$$

$$X_{U1t} = 0 + (0.2509)\cdot \tan(68.438) = 0.5210$$

$$\rho_{t1} = \frac{14.69}{53.35\cdot 517.8}\cdot 144 = 0.07658 \text{ lbm/ft}^3$$

$$V_{t1} = \sqrt{2\cdot 32.2\cdot 186.725\cdot 517.8} = 2495.3139 \text{ ft/sec}$$

$$\Phi_{1t} = 0.2059\cdot\left(1-0.2059^2\right)^{\frac{1}{1.4-1}} = 0.1847$$

$$A_1 = \left(\frac{17.09}{0.07658\cdot 2495.3139}\right)\cdot\frac{1}{0.1847\cdot\cos 0} = 0.4841 \text{ ft}^2 \text{ or } 69.7144 \text{ in}^2$$

$$r_{1t} = \frac{0.5210\cdot 2495.3139}{2836.33} = 0.4583 \text{ ft or } 5.5 \text{ in}$$

$$r_{1h} = \sqrt{5.5^2 - 69.714/\pi} = 2.8389 \text{ in or } 0.2366 \text{ ft}$$

$$r_{1ht} = 2.8389/5.5 = 0.5162$$

$$r_{1m} = \frac{5.5+2.8389}{2} = 4.1695 \text{ in or } 0.3475 \text{ ft}$$

$$X_{\theta 1m} = \frac{5.5}{4.1695}\cdot 0 = 0$$

$$X_{Z1m} = X_{Z1t} = 0.2058$$

$$\alpha_{1m} = \tan^{-1}\left(\frac{5.5}{4.1695}\cdot\tan 0\right) = 0$$

$$X_{U1m} = \frac{4.1695}{5.5}\cdot 0.5210 = 0.3949$$

$$\beta_{1m} = \tan^{-1}\left(\frac{0.3949-0}{0.2058}\right) = 62.4685^\circ$$

$$X_{1m} = \frac{0.2058}{\cos 0} = 0.2058$$

$$M_{1m} = \sqrt{\frac{\frac{2}{1.4-1}\cdot 0.2058^2}{1-0.2058^2}} = 0.4704$$

$$X_{W1m} = \frac{0.2058}{\cos 62.4685} = 0.4454$$

$$M_{W1m} = \sqrt{\frac{\frac{2}{1.4-1}\cdot 0.4454^2}{1-0.4454^2}} = 1.1123$$

2. Rotor Conditions at Mean Line

$$0.455 = 1-(0.956)\cdot\frac{\cos 62.4685}{\cos\beta_{2m}} + \frac{\tan 62.4685 - 1.0687\cdot 0.956\cdot\tan\beta_{2m}}{(1+1.0687)\cdot 1.52}\cdot\cos 62.4685$$

Solve for β_{2m}

$$\sin\beta_{2m} = \frac{-B+\sqrt{B^2+4\left(R_{21}^2+A^2\right)\cdot\left[A^2-B^2\big/4\cdot R_{21}^2\right]}}{2\cdot\left(R_{21}^2+A^2\right)}$$

$$\text{Where } A = \left(\left(1+R_{21}\right)\cdot\sigma_{2m}\cdot\left[\frac{1-D_{2m}}{\phi_{21}\cdot\cos\beta_{1m}}\right]+\frac{\tan\beta_{1m}}{\phi_{21}}\right)$$

$$A = \left(\left(1+1.0687\right)\cdot 1.52\cdot\left[\frac{1-0.455}{0.956\cdot\cos 62.4685}\right]+\frac{\tan 62.4685}{0.956}\right) = 5.8848$$

$$\text{and } B = 2\cdot\sigma_{2m}\cdot R_{21}\cdot\left(1+R_{21}\right) = 2\cdot 1.52\cdot 1.0687\cdot\left(1+1.0687\right) = 6.7209$$

$$\therefore\quad \sin\beta_{2m} = \frac{-6.7209+\sqrt{6.7209^2+4\left(1.0687^2+5.8848^2\right)\cdot\left[5.8848^2-\frac{6.7209^2}{4\cdot 1.0687^2}\right]}}{2\cdot\left(1.0687^2+5.8848^2\right)}$$

$$\sin\beta_{2m} = 0.7430$$

$$\beta_{2m} = 47.9895^\circ$$

$$X_{U2m} = 1.0687\cdot 0.3949 = 0.4221$$

$$\phi_{1m} = \frac{0.2059}{0.3949} = 0.5213$$

$$\phi_{2m} = 0.956\cdot 0.5213\cdot\left(\frac{1}{1.0687}\right) = 0.4663$$

$$X_{Z2m} = 0.4663\cdot 0.4213 = 0.1968$$

$$X_{\theta 2m} = 0.4221-0.1968\tan 47.9895 = 0.2036$$

$$\alpha_{2m} = \tan^{-1}\left(\frac{0.2036}{0.1968}\right) = 45.9677^{\circ}$$

$$r_{st_m} = \left[1 - \frac{1}{2}\left(\frac{0.0}{0.3949} + \frac{0.2036}{0.4221}\right)\right] = 0.7588$$

$$\tau = 1 + 2 \cdot [0.4221 \cdot 0.2036 - 0.3949 \cdot 0.0] = 1.1718$$

$$X_{2m} = \frac{0.1968}{\cos(45.9677)} = 0.2832$$

$$X_{W2m} = \frac{0.1968}{\cos(47.9894)} = 0.2941$$

$$Y_{2m} = \frac{0.2832}{\sqrt{1.1718}} = 0.2616$$

$$Y_{W2m} = \frac{0.2941}{\sqrt{1.1718}} = 0.2717$$

$$M_{2m} = \sqrt{\frac{\frac{2}{1.4-1} \cdot 0.2616^2}{1 - 0.2616^2}} = 0.6060$$

$$M_{W2m} = \sqrt{\frac{\frac{2}{1.4-1} \cdot 0.2717^2}{1 - 0.2717^2}} = 0.6312$$

3. Stator Conditions at Mean Line

$$\alpha_{3m} = 1 \cdot 0 = 0^{\circ}$$

$$D_{3m} = 1 - 1.0371 \cdot \frac{\cos 45.9677}{\cos 0} + \frac{(\tan 45.9677 - 1.0259 \cdot 1.0371 \cdot \tan 0) \cdot \cos 45.9677}{(1 + 1.0259) \cdot 1.25} = 0.5631$$

$$X_{Z3m} = 1.0371 \cdot 0.1968 = 0.2041$$

$$X_{\theta 3m} = 0.2041 \cdot \tan 0 = 0$$

$$X_{U3m} = 0.4221 \cdot 1.0259 = 0.4330$$

$$\beta_{3m} = \tan^{-1}\left(\frac{0.4330 - 0}{0.2041}\right) = 64.7615^{\circ}$$

$$X_{3m} = \frac{0.2041}{\cos 0} = 0.2041$$

$$X_{W3m} = \frac{0.2041}{\cos 64.7615} = 0.4787$$

$$Y_{3m} = \frac{0.2041}{\sqrt{1.1718}} = 0.1886$$

$$Y_{W3m} = \frac{0.4787}{\sqrt{1.1718}} = 0.4422$$

$$M_{3m} = \sqrt{\frac{\frac{2}{1.4-1}\cdot 0.1885^2}{1-0.1885^2}} = 0.4293$$

$$M_{W3m} = \sqrt{\frac{\frac{2}{1.4-1}\cdot 0.4422^2}{1-0.4422^2}} = 1.1025$$

4. Iteration #1

Rotor Performance

Initially set $\tilde{\omega}_{S_2}$ and $\tilde{\omega}_{SFTC_2} = 0$

$$\tilde{\omega}_{P_2} = 2\cdot 1.52\cdot\frac{\cos^2 62.4685}{\cos^3 47.9894}\cdot\left[0.005+0.16\cdot 0.455^4\right] = 0.02569$$

$$\tilde{\omega}_{T_2} = 0+0+0.02569$$

$$T_1/T_{t1} = 1-0.2059^2 = 0.9576$$

$$P_1/P_{t1} = (0.9576)^{3.5} = 0.8594$$

$$T_{R1}/T_{t1} = 0.9576+0.4454^2 = 1.1560$$

$$P_{R1}/P_{t1} = (1.1560)^{3.5} = 1.6608$$

$$T_{E1}/T_{t1} = 1.1560+0.4221^2-0.3949^2 = 1.1781$$

$$T_{E2}/T_{t1} = 1.1781$$

$$P_{E1}/P_{t1} = (1.1781)^{3.5} = 1.7750$$

$$P_{E2}/P_{t1} = 1.7750-0.02569\cdot[1.6608-0.8594] = 1.7544$$

$$P_{t2}/P_{t1} = 1.7544\cdot\left[\frac{1.1718}{1.1781}\right]^{3.5} = 1.7218$$

$$T_2/T_{t1} = 1.1718-0.2832^2 = 1.0917$$

$$P_2/P_{t1} = 1.7218\cdot\left(\frac{1.0917}{1.1718}\right)^{3.5} = 1.3435$$

$$\Phi_{2m} = 0.2616 \cdot \left(1 - 0.2616^2\right)^{2.5} = 0.2191$$

$$A_2 / A_1 = \frac{0.1847 \cdot \cos 0}{0.2191 \cdot \cos 45.9677} \cdot \frac{\sqrt{1.1718}}{1.7218} = 0.7627$$

$$A_2 = 0.7627 \cdot 69.7144 = 53.1725 \text{ in}^2 \text{ or } 0.3693 \text{ ft}^2$$

$$r_{2m} = 4.1694 \cdot 1.0687 = 4.4559 \text{ in or } 0.3713 \text{ ft}$$

$$H_2 = \frac{53.1725}{2 \cdot \pi \cdot 4.4559} = 1.8992 \text{ in or } 0.1583 \text{ ft}$$

$$r_{ht2} = \frac{1 - \dfrac{1.8992}{2 \cdot 4.4559}}{1 + \dfrac{1.8992}{2 \cdot 4.4559}} = 0.6487$$

$$r_{2t} = \left(\frac{2}{1 + 0.6487}\right) \cdot 4.4559 = 5.4055 \text{ in or } 0.4505 \text{ ft}$$

$$r_{2h} = 0.6487 \cdot 5.4055 = 3.5063 \text{ in or } 0.2922 \text{ ft}$$

Stator Performance

Initially set $\tilde{\omega}_{S_3}$ and $\tilde{\omega}_{SFTC_3} = 0$

$$\tilde{\omega}_{P_3} = 21.25 \cdot \frac{\cos^2 45.9677}{\cos^3 0} \cdot \left[0.005 + 0.16 \cdot 0.5631^4\right] = 0.02546$$

$$\tilde{\omega}_{T_3} = 0 + 0 + 0.02546$$

$$P_{t3} / P_{t1} = 1.7218 - 0.02546 \cdot \left(1.7218 - 1.3435\right) = 1.7121$$

$$T_{t3} / T_{t1} = \tau = 1.1718$$

$$T_3 / T_{t1} = 1.1718 - 0.2041^2 = 1.1302$$

$$P_3 / P_{t1} = \left(\frac{1.1302}{1.1718}\right)^{3.5} = 1.5084$$

$$\Phi_{3m} = 0.1886 \cdot \left(1 - 0.1886^2\right)^{2.5} = 0.1722$$

$$A_3 / A_1 = \frac{0.1847 \cdot \cos 0}{0.1722 \cdot \cos 0} \cdot \frac{\sqrt{1.1718}}{1.7121} = 0.6782$$

$$A_3 = 0.6782 \cdot 69.7144 = 47.2778 \text{ in}^2 \text{ or } 0.3233 \text{ ft}^2$$

$$r_{3m} = 1.0259 \cdot 4.4559 = 4.5713 \text{ in or } 0.3809 \text{ ft}$$

$$H_3 = \frac{47.2778}{2 \cdot \pi \cdot 4.5713} = 1.6460 \text{ in or } 0.1372 \text{ ft}$$

$$r_{ht3} = \frac{1 - \dfrac{1.6460}{2 \cdot 4.5713}}{1 + \dfrac{1.6460}{2 \cdot 4.5713}} = 0.6949$$

$$r_{3t} = \left(\frac{2}{1 + 0.6949} \right) \cdot 4.5713 = 5.3943 \text{ in or } 0.4495 \text{ ft}$$

$$r_{3h} = 0.6949 \cdot 5.3943 = 3.7483 \text{ in or } 0.3124 \text{ ft}$$

Rotor Hub Calculations

$$X_{\theta 2h} = 0.2036 \cdot \frac{4.4559}{3.5063} = 0.2587$$

$$X_{U2h} = 0.4221 \cdot \frac{3.5063}{4.4559} = 0.3321$$

$$\alpha_{2h} = \tan^{-1}\left(\frac{0.2587}{0.1968} \right) = 52.7381^{\circ}$$

$$\beta_{2h} = \tan^{-1}\left(\frac{0.3321 - 0.2587}{0.1968} \right) = 20.4569^{\circ}$$

$$X_{2h} = \frac{0.1968}{\cos 52.7381} = 0.3251$$

$$X_{W2h} = \frac{0.1968}{\cos 20.4569} = 0.2101$$

$$Y_{2h} = \frac{0.3251}{\sqrt{1.1718}} = 0.3003$$

$$Y_{W2h} = \frac{0.2101}{\sqrt{1.1718}} = 0.1940$$

$$M_{2h} = \sqrt{\frac{\dfrac{2}{1.4 - 1} \cdot 0.3003^2}{1 - 0.3003^2}} = 0.7039$$

$$M_{W2h} = \sqrt{\frac{\dfrac{2}{1.4 - 1} \cdot 0.1940^2}{1 - 0.1940^2}} = 0.4423$$

Rotor Tip Calculations

$$X_{\theta 2t} = 0.2036 \cdot \frac{4.4559}{5.4055} = 0.1678$$

$$X_{U2t} = 0.4221 \cdot \frac{5.4055}{4.4559} = 0.5120$$

$$\alpha_{2t} = \tan^{-1}\left(\frac{0.1678}{0.1968}\right) = 40.4526^{\circ}$$

$$\beta_{2t} = \tan^{-1}\left(\frac{0.5120 - 0.1678}{0.1968}\right) = 60.2402^{\circ}$$

$$X_{2t} = \frac{0.1968}{\cos 40.4526} = 0.2586$$

$$X_{W2t} = \frac{0.1968}{\cos 60.2402} = 0.3965$$

$$Y_{2t} = \frac{0.2586}{\sqrt{1.1718}} = 0.2389$$

$$Y_{W2t} = \frac{0.3965}{\sqrt{1.1718}} = 0.3663$$

$$M_{2t} = \sqrt{\frac{\frac{2}{1.4-1}\cdot 0.2389^2}{1-0.2389^2}} = 0.5502$$

$$M_{W2t} = \sqrt{\frac{\frac{2}{1.4-1}\cdot 0.3663^2}{1-0.3663^2}} = 0.8802$$

Stator Hub Calculations

$$X_{\theta 3h} = 0.0 \cdot \frac{4.5713}{3.7483} = 0.0$$

$$X_{U3h} = 0.4330 \cdot \frac{3.7483}{4.5713} = 0.3550$$

$$\alpha_{3h} = \tan^{-1}\left(\frac{0.0}{0.2041}\right) = 0.0^{\circ}$$

$$\beta_{3h} = \tan^{-1}\left(\frac{0.3550 - 0.0}{0.2041}\right) = 60.1058^{\circ}$$

$$X_{3h} = \frac{0.2041}{\cos 0} = 0.2041$$

$$X_{W3h} = \frac{0.2041}{\cos 60.1058} = 0.4095$$

$$Y_{3h} = \frac{0.2041}{\sqrt{1.1718}} = 0.1886$$

$$Y_{W3h} = \frac{0.4095}{\sqrt{1.1718}} = 0.3783$$

$$M_{3h} = \sqrt{\frac{\frac{2}{1.4-1}\cdot 0.1886^2}{1-0.1886^2}} = 0.4293$$

$$M_{W3h} = \sqrt{\frac{\frac{2}{1.4-1}\cdot 0.3783^2}{1-0.3783^2}} = 0.9139$$

Stator Tip Calculations

$$X_{\theta 3t} = 0.0\cdot\frac{4.5713}{5.3943} = 0.0$$

$$X_{U3t} = 0.4330\cdot\frac{5.3943}{4.5713} = 0.5110$$

$$\alpha_{3t} = \tan^{-1}\left(\frac{0.0}{0.2041}\right) = 0.0^\circ$$

$$\beta_{3t} = \tan^{-1}\left(\frac{0.5110-0.0}{0.2041}\right) = 68.2250^\circ$$

$$X_{3t} = \frac{0.2041}{\cos 0} = 0.2041$$

$$X_{W3t} = \frac{0.2041}{\cos 68.2250} = 0.5502$$

$$Y_{3t} = \frac{0.2041}{\sqrt{1.1718}} = 0.1886$$

$$Y_{W3t} = \frac{0.5502}{\sqrt{1.1718}} = 0.5083$$

$$M_{3t} = \sqrt{\frac{\frac{2}{1.4-1}\cdot 0.1886^2}{1-0.1886^2}} = 0.4293$$

$$M_{W3t} = \sqrt{\frac{\frac{2}{1.4-1}\cdot 0.5083^2}{1-0.5083^2}} = 1.3197$$

Rotor Blade Geometry

$$H_2 = 5.4055 - 3.5063 = 1.8992 \text{ in}$$

$$C_2' = \frac{1.8992}{1.2} = 1.5827 \text{ in}$$

$$Z_2' = \frac{2\cdot\pi\cdot 4.559}{1.5827}\cdot 1.52 = 26.8884 \text{ blades}$$

Choose Z_2 to match transonic compressor rotor. Therefore, $Z_2 = 22$.

$$AR_{Rev2} = \frac{1.8992}{\frac{2 \cdot \pi \cdot 4.559 \cdot 1.52}{22}} = 0.9818$$

$$C_{Rev2} = \frac{1.8992}{0.9818} = 1.9343 \text{ in}$$

$$S_2 = \frac{1.9343}{1.52} = 1.2726 \text{ in}$$

Stator Blade Geometry

$$H_3 = 5.3943 - 3.7483 = 1.6460 \text{ in}$$

$$C_3' = \frac{1.6460}{1.2} = 1.3717 \text{ in}$$

$$Z_3' = \frac{2 \cdot \pi \cdot 4.5713}{1.3717} \cdot 1.25 = 26.1740 \text{ blades}$$

Choose Z_3 to match transonic compressor rotor. Therefore, $Z_3 = 27$.

$$AR_{Rev3} = \frac{1.6460}{\frac{2 \cdot \pi \cdot 4.5713 \cdot 1.25}{27}} = 1.2379$$

$$C_{Rev3} = \frac{1.6460}{1.2379} = 1.3297 \text{ in}$$

$$S_3 = \frac{1.3297}{1.25} = 1.0638 \text{ in}$$

The design calculations have to iterated until the losses converge to a chosen criteria. A criterion of 0.0001 was chosen for both the hand and computer calculations.

5. Iteration #2

Only the results will be shown for all subsequent iterations.

ROTOR		STATOR	
β_∞	56.560435987322300	β_∞	27.347157421794000
C_L	0.586010932091710	C_L	1.470018487969480
C_{Di}	0.014452454402386	C_{Di}	0.071059015271782
ω_{SFTC2}	0.028049370920909	ω_{SFTC3}	0.061233413235418
y*	0.908770674133602	y*	1.978417070245130
P_{te}/P_{ti}	0.999746596870648	P_{te}/P_{ti}	1.005788072708940
ω_{S2}	0.000470350655165	ω_{S3}	-0.026347850522881
ω_{P2}	0.025691775310535	ω_{P3}	0.025461335476916
ω_{T2}	0.054211496886609	ω_{T3}	0.086694748712334
T_1/T_{t1}	0.957618541339988	P_{t3}/P_{t1}	1.666977507638590
P_1/P_{t1}	0.859357579736388	T_{t3}/T_{t1}	1.171843486972510
T_{R1}/T_{t1}	1.15597584087194	T_3/T_{t1}	1.130182173979650

P_{R1}/P_{t1}	1.660812680211420	P_3/P_{t1}	1.468607319851730
T_{E1}/T_{t1}	1.178143079024150	Φ_{3m}	0.172237922031280
P_{E1}/P_{t1}	1.774978623835910	A_3/A_1	0.696539265527826
P_{E2}/P_{t1}	1.731530543151750	A_3 (in)	48.558787505362500
P_{t2}/P_{t1}	1.699341511605150	r_{3m} (in)	4.571286972682070
T_2/T_{t1}	1.091667880043820	h_3 (in)	1.690633536728720
P_2/P_{t1}	1.326031657810530	r_{ht3}	0.687879413456575
Φ_{2m}	0.219098107941022	r_{3t} (in)	5.416603741046430
A_2/A_1	0.772787543503894	r_{3h} (in)	3.725970204317710
A_2 (in)	53.874387229785800		
r_{2m} (in)	4.455879688743610		
h_2 (in)	1.924283336313540		
r_{ht2}	0.644836516957841		
r_{2t} (in)	5.418021356900380		
r_{2h} (in)	3.493738020586840		

Table C1. Stage Performance Results (Iteration #2)

HUB		**TIP**	
$X_{\theta 2h}$	0.259634095360950	$X_{\theta 2t}$	0.167421545736055
X_{U2h}	0.330933976012678	X_{U2t}	0.513206010065825
α_{2h}	52.836925539652200	α_{2t}	40.387101220022500
β_{2h}	19.914379724264200	β_{2t}	60.352859069903200
X_{2h}	0.325797182731520	X_{2t}	0.258387141279412
X_{W2h}	0.209326572089755	X_{W2t}	0.397870376597357
Y_{2h}	0.300962592479341	Y_{2t}	0.238691026272198
Y_{W2h}	0.193370204379153	Y_{W2t}	0.367541852288360
M_{2h}	0.705691535783612	M_{2t}	0.549615692085776
M_{W2h}	0.440706882771685	M_{W2t}	0.883701501259568
M_{Z2h}	0.426298128621960	M_{Z2t}	0.418633587082212

Table C2. Rotor Results (Iteration #2)

HUB		**TIP**	
$X_{\theta 3h}$	0.000000000000000	$X_{\theta 3t}$	0.000000000000000
X_{U3h}	0.352931481111029	X_{U3t}	0.513071730606907
α_{3h}	0.000000000000000	α_{3t}	0.000000000000000
β_{3h}	59.957896644744600	β_{3t}	68.306267419874700
X_{3h}	0.204111031041579	X_{3t}	0.204111031041579
X_{W3h}	0.407703254036660	X_{W3t}	0.552181051595238
Y_{3h}	0.188552229153336	Y_{3t}	0.188552229153336
Y_{W3h}	0.376625197518214	Y_{W3t}	0.510089864537044

M_{3h}	0.429316175796641	M_{3t}	0.429316175796641
M_{W3h}	0.909100497709792	M_{W3t}	1.326087214551410
M_{Z3h}	0.429316175796641	M_{Z3t}	0.429316175796641

Table C3. Stator Results (Iteration #2)

ROTOR		STATOR	
H_2	1.924283336313540	H_3	1.690633536728720
C_2'	1.603569446927950	C_3'	1.408861280607270
Z_2'	26.538058032755800	Z_3'	25.483562094627500
Z_2	22.000000000000000	Z_3	27.000000000000000
AR'	0.994797734160298	AR'	1.271407814954980
C_2	1.934346320096740	C_3	1.329733478780440
S_2	1.272596263221540	S_3	1.063786783024350

Table C4. Blade Geometry Results (Iteration #2)

Compare $\tilde{\omega}_T$ (iteration #2) and $\tilde{\omega}_T$ (iteration #1).

$$\left|\tilde{\omega}_{T2_{iteration2}} - \tilde{\omega}_{T2_{iteration1}}\right| = 0.05421 - 0.02569 = 0.02852$$

$$\left|\tilde{\omega}_{T3_{iteration2}} - \tilde{\omega}_{T3_{iteration1}}\right| = 0.08669 - 0.02546 = 0.06123$$

6. Iteration #3

ROTOR		STATOR	
$\beta\infty$	56.560435987322300	$\beta\infty$	27.347157421794000
C_L	0.586010932091710	C_L	1.470018487969480
C_{Di}	0.014264158667470	C_{Di}	0.069184495778439
ω_{SFTC2}	0.027683925940807	ω_{SFTC3}	0.059618090727575
y*	0.908770674133602	y*	1.978417070245130
P_{te}/P_{ti}	0.999746596870648	P_{te}/P_{ti}	1.005788072708940
ω_{S2}	0.000470350655165	ω_{S3}	-0.026347850522881
ω_{P2}	0.025691775310535	ω_{P3}	0.025461335476916
ω_{T2}	0.053846051906507	ω_{T3}	0.085079426204491
T_1/T_{t1}	0.957618541339988	P_{t3}/P_{t1}	1.667862594069170
P_1/P_{t1}	0.859357579736388	T_{t3}/T_{t1}	1.171843486972510
T_{R1}/T_{t1}	1.15597584087194	T_3/T_{t1}	1.130182173979650
P_{R1}/P_{t1}	1.660812680211420	P_3/P_{t1}	1.469387081069080
T_{E1}/T_{t1}	1.178143079024150	Φ_{3m}	0.172237922031280
P_{E1}/P_{t1}	1.774978623835910	A_3/A_1	0.696169632289165
P_{E2}/P_{t1}	1.731823430894990	A_3 (in)	48.533018761546700
P_{t2}/P_{t1}	1.699628954585760	r_{3m} (in)	4.571286972682070
T_2/T_{t1}	1.091667880043820	h_3 (in)	1.689736366417590
P_2/P_{t1}	1.326255955569100	r_{ht3}	0.688019209608545
Φ_{2m}	0.219098107941022	r_{3t} (in)	5.416155155890860
A_2/A_1	0.772656848887118	r_{3h} (in)	3.726418789473280

A_2 (in)	53.865275938523100		
r_{2m} (in)	4.455879688743610		
h_2 (in)	1.923957899554990		
r_{ht2}	0.644885917495964		
r_{2t} (in)	5.417858638521100		
r_{2h} (in)	3.493900738966110		

Table C5. Stage Performance Results (Iteration #3)

HUB		TIP	
$X_{\theta 2h}$	0.259622003649605	$X_{\theta 2t}$	0.167426574025716
X_{U2h}	0.330949389028739	X_{U2t}	0.513190597049764
α_{2h}	52.835640920353900	α_{2t}	40.387950481631600
β_{2h}	19.921457676992400	β_{2t}	60.351402895843300
X_{2h}	0.325787546694057	X_{2t}	0.258390399380250
X_{W2h}	0.209335942224263	X_{W2t}	0.397852611427487
Y_{2h}	0.300953690969537	Y_{2t}	0.238694036017300
Y_{W2h}	0.193378860255026	Y_{W2t}	0.367525441306733
M_{2h}	0.705668584925989	M_{2t}	0.549623041103662
M_{W2h}	0.440727376594600	M_{W2t}	0.883655881142935
M_{Z2h}	0.426296872834759	M_{Z2t}	0.418633906000314

Table C6. Rotor Results (Iteration #3)

HUB		TIP	
$X_{\theta 3h}$	0.000000000000000	$X_{\theta 3t}$	0.000000000000000
X_{U3h}	0.352973972009957	X_{U3t}	0.513029239707979
α_{3h}	0.000000000000000	α_{3t}	0.000000000000000
β_{3h}	59.960885862080200	β_{3t}	68.304637549438200
X_{3h}	0.204111031041579	X_{3t}	0.204111031041579
X_{W3h}	0.407740037167485	X_{W3t}	0.552141570422119
Y_{3h}	0.188552229153336	Y_{3t}	0.188552229153336
Y_{W3h}	0.376659176775861	Y_{W3t}	0.510053392901174
M_{3h}	0.429316175796641	M_{3t}	0.429316175796641
M_{W3h}	0.909196076420228	M_{W3t}	1.325959056752780
M_{Z3h}	0.429316175796641	M_{Z3t}	0.429316175796641

Table C7. Stator Results (Iteration #3)

ROTOR		STATOR	
H_2	1.923957899554990	H_3	1.689736366417590
C_2'	1.603298249629160	C_3'	1.408113638681320

Z_2'	26.542546935338600	Z_3'	25.497092664122100
Z_2	22.000000000000000	Z_3	27.000000000000000
AR'	0.994629492953864	AR'	1.270733115606990
C_2	1.934346320096740	C_3	1.329733478780440
S_2	1.272596263221540	S_3	1.063786783024350

Table C8. Blade Geometry Results (Iteration #3)

Compare $\tilde{\omega}_T$ (iteration #3) and $\tilde{\omega}_T$ (iteration #2).

$$\left|\tilde{\omega}_{T2_{iteration3}} - \tilde{\omega}_{T2_{iteration2}}\right| = 0.05385 - 0.05421 = 0.0003654$$

$$\left|\tilde{\omega}_{T3_{iteration3}} - \tilde{\omega}_{T3_{iteration2}}\right| = 0.08508 - 0.08669 = 0.001615$$

7. Iteration #4

ROTOR		STATOR	
β_∞	56.560435987322300	β_∞	27.347157421794000
C_L	0.586010932091710	C_L	1.470018487969480
C_{Di}	0.014266571444569	C_{Di}	0.069221229482487
ω_{SFTC2}	0.027688608666518	ω_{SFTC3}	0.059649745121757
y*	0.908770674133602	y*	1.978417070245130
P_{te}/P_{ti}	0.999746596870648	P_{te}/P_{ti}	1.005788072708940
ω_{S2}	0.000470350655165	ω_{S3}	-0.026347850522881
ω_{P2}	0.025691775310535	ω_{P3}	0.025461335476916
ω_{T2}	0.053850734632218	ω_{T3}	0.085111080598674
T_1/T_{t1}	0.957618541339988	P_{t3}/P_{t1}	1.667847160812340
P_1/P_{t1}	0.859357579736388	T_{t3}/T_{t1}	1.171843486972510
T_{R1}/T_{t1}	1.15597584087194	T_3/T_{t1}	1.130182173979650
P_{R1}/P_{t1}	1.660812680211420	P_3/P_{t1}	1.469373484368560
T_{E1}/T_{t1}	1.178143079024150	Φ_{3m}	0.172237922031280
P_{E1}/P_{t1}	1.774978623835910	A_3/A_1	0.696176074225205
P_{E2}/P_{t1}	1.731819677900590	A_3 (in)	48.533467856980100
P_{t2}/P_{t1}	1.699625271359260	r_{3m} (in)	4.571286972682070
T_2/T_{t1}	1.091667880043820	h_3 (in)	1.689752002224000
P_2/P_{t1}	1.326253081470570	r_{ht3}	0.688016773055056
Φ_{2m}	0.219098107941022	r_{3t} (in)	5.416162973794070
A_2/A_1	0.772658523297490	r_{3h} (in)	3.726410971570070
A_2 (in)	53.865392668966700		
r_{2m} (in)	4.455879688743610		
h_2 (in)	1.923962068928580		
r_{ht2}	0.644885284576048		
r_{2t} (in)	5.417860723207900		
r_{2h} (in)	3.493898654279320		

Table C9. Stage Performance Results (Iteration #4)

HUB		TIP	
$X_{\theta 2h}$	0.259622158556950	$X_{\theta 2t}$	0.167426509603247
X_{U2h}	0.330949191562962	X_{U2t}	0.513190794515540
α_{2h}	52.835657378138000	α_{2t}	40.387939601023500
β_{2h}	19.921367002758800	β_{2t}	60.351421552757200
X_{2h}	0.325787670140640	X_{2t}	0.258390357637089
X_{W2h}	0.209335822159843	X_{W2t}	0.397852839028210
Y_{2h}	0.300953805006140	Y_{2t}	0.238693997456101
Y_{W2h}	0.193378749342774	Y_{W2t}	0.367525651558102
M_{2h}	0.705668878946494	M_{2t}	0.549622946947096
M_{W2h}	0.440727113995883	M_{W2t}	0.883656465604799
M_{Z2h}	0.426296888922253	M_{Z2t}	0.418633901914269

Table C10. Rotor Results (Iteration #4)

HUB		TIP	
$X_{\theta 3h}$	0.000000000000000	$X_{\theta 3t}$	0.000000000000000
X_{U3h}	0.352973231482253	X_{U3t}	0.513029980235682
α_{3h}	0.000000000000000	α_{3t}	0.000000000000000
β_{3h}	59.960833770877400	β_{3t}	68.304665956674600
X_{3h}	0.204111031041579	X_{3t}	0.204111031041579
X_{W3h}	0.407739396104768	X_{W3t}	0.552142258492756
Y_{3h}	0.188552229153336	Y_{3t}	0.188552229153336
Y_{W3h}	0.376658584579527	Y_{W3t}	0.510054028522148
M_{3h}	0.429316175796641	M_{3t}	0.429316175796641
M_{W3h}	0.909194410621443	M_{W3t}	1.325961290180600
M_{Z3h}	0.429316175796641	M_{Z3t}	0.429316175796641

Table C11. Stator Results (Iteration #4)

ROTOR		STATOR	
H_2	1.923962068928580	H_3	1.689752002224000
C_2'	1.603301724107150	C_3'	1.408126668520000
Z_2'	26.542489415600500	Z_3'	25.496856731509300
Z_2	22.000000000000000	Z_3	27.000000000000000
AR'	0.994631648397049	AR'	1.270744874208740
C_2	1.934346320096740	C_3	1.329733478780440
S_2	1.272596263221540	S_3	1.063786783024350

Table C12. Blade Geometry Results (Iteration #4)

Compare $\tilde{\omega}_T$ (iteration #4) and $\tilde{\omega}_T$ (iteration #3).

$$\left|\tilde{\omega}_{T2_{iteration4}} - \tilde{\omega}_{T2_{iteration3}}\right| = 0.05385 - 0.05385 = 0.000004683$$

$$\left|\tilde{\omega}_{T3_{iteration4}} - \tilde{\omega}_{T3_{iteration3}}\right| = 0.08511 - 0.08508 = 0.00003165$$

Both the rotor and stator losses satisfy the criterion after the fourth iteration.

$$\Pi_C = \frac{P_{t3}}{P_{t1}} = 1.6678$$

$$\tau_C = \frac{T_{t2}}{T_{t1}} = 1.1718$$

$$\eta_C = \frac{1.6678^{\frac{1.4-1}{1.4}} - 1}{1.1718 - 1} = 0.9158 \text{ or } 91.58\%$$

C2. CODE RESULTS

HUB		MEAN		TIP	
β_{1h}	52.195176449181800	β_{1m}	62.343578122316700	β_{1t}	68.410458367312200
α_{1h}	0.000000000000000	α_{1m}	0.000000000000000	α_{1t}	0.000000000000000
X_{1h}	0.205867575543143	X_{1m}	0.205867575543143	X_{1t}	0.205867575543143
X_{U1h}	0.265557741712975	X_{U1m}	0.393264467850964	X_{U1t}	0.520971193988953
r_{1h}	2.794827442941060	r_{1m}	4.138860046006270	r_{1t}	5.482892649071480
$X_{\theta 1h}$	0.000000000000000	$X_{\theta 1m}$	0.000000000000000	$X_{\theta 1t}$	0.000000000000000
X_{Z1h}	0.205867575543143	X_{Z1m}	0.205867575543143	X_{Z1t}	0.205867575543143
M_{Z1h}	0.470410144688393	M_{Z1m}	0.470410144688393	M_{Z1t}	0.470410144688393
X_{W1h}	0.336009483264546	X_{W1m}	0.443890076859253	X_{W1t}	0.560171798313952
M_{1h}	0.470410144688393	M_{1m}	0.470410144688393	M_{1t}	0.470410144688393
M_{W1h}	0.797720638371130	M_{W1m}	1.107676776722660	M_{W1t}	1.280000000000000
		A_1	69.931875719909200		
		r_{ht1}	0.509735940829400		
		ω	2836.334567415990000		
		mdot	17.090000000000000		
		P_{t1}	14.690000000000000		
		T_{t1}	517.800000000000000		
		R	53.350000000000000		
		γ	1.400000000000000		

Table C13. Inlet Results

HUB		MEAN		TIP	
β_{2h}	19.153113641505600	β_{2m}	47.808100398602100	β_{2t}	60.322721543344400
α_{2h}	52.844869054566300	α_{2m}	45.871861712689900	α_{2t}	40.226189216790300
X_{2h}	0.326016514230627	X_{2m}	0.282758297008589	X_{2t}	0.257833958315095
X_{U2h}	0.328294695162428	X_{U2m}	0.420281736792325	X_{U2t}	0.512268778422222
r_{2h}	3.455094238614300	r_{2m}	4.423199731166900	r_{2t}	5.391305223719500
$X_{\theta 2h}$	0.259909266378080	$X_{\theta 2m}$	0.203022938914057	$X_{\theta 2t}$	0.166566531027495

X_{Z2h}	0.196809402219245	X_{Z2m}	0.196809402219245	X_{Z2t}	0.196809402219245
M_{Z2h}	0.426564947028765	M_{Z2m}	0.421383869310173	M_{Z2t}	0.421383869310173
X_{W2h}	0.208351884253337	X_{W2m}	0.293147277076569	X_{W2t}	0.397798924905110
M_{2h}	0.706608604848915	M_{2m}	0.605406977153977	M_{2t}	0.548663539641709
M_{W2h}	0.438807084275359	M_{W2m}	0.629379184024538	M_{W2t}	0.629379184024538
Y_{2h}	0.301318214114674	Y_{2m}	0.261337145088495	Y_{2t}	0.238301019937484
Y_{W2h}	0.192567599892291	Y_{W2m}	0.270939078683619	Y_{W2t}	0.367662545905120
D_{2h}	0.539065878334816	D_{2m}	0.455000000000000	D_{2t}	0.390436898079840
r_{st2h}	0.604152503515955	r_{st2m}	0.758468045193244	r_{st2t}	0.837422718264700
		τ	1.170653666750960		
		r_{ht2}	0.640864149819104		
		σ_{2m}	1.520000000000000		
		$f\sigma_2$	0.855300000000000		
		R_{21}	1.068700000000000		
		ϕ_{21}	0.956000000000000		

Table C14. Rotor Results

HUB		MEAN		TIP	
β_{3h}	59.760648493304300	β_{3m}	64.641480976070200	β_{3t}	68.231904833353900
α_{3h}	0.000000000000000	α_{3m}	0.000000000000000	α_{3t}	0.000000000000000
X_{3h}	0.204111031041579	X_{3m}	0.204111031041579	X_{3t}	0.204111031041579
X_{U3h}	0.350482583299427	X_{U3m}	0.431167033775246	X_{U3t}	0.511851484251066
r_{3h}	3.688607742179240	r_{3m}	4.537760604204120	r_{3t}	5.386913466229010
$X_{\theta 3h}$	0.000000000000000	$X_{\theta 3m}$	0.000000000000000	$X_{\theta 3t}$	0.000000000000000
X_{Z3h}	0.204111031041579	X_{Z3m}	0.204111031041579	X_{Z3t}	0.204111031041579
M_{Z3h}	0.429542339664783	M_{Z3m}	0.429542339664783	M_{Z3t}	0.429542339664783
X_{W3h}	0.405585199667217	X_{W3m}	0.477039122093147	X_{W3t}	0.551047416220125
M_{3h}	0.429542339664783	M_{3m}	0.429542339664783	M_{3t}	0.429542339664783
M_{W3h}	0.904138170145811	M_{W3m}	1.098406365761990	M_{W3t}	1.098406365761990
Y_{3h}	0.188648024471074	Y_{3m}	0.188648024471074	Y_{3t}	0.188648024471074
Y_{W3h}	0.374858949472163	Y_{W3m}	0.440899678567374	Y_{W3t}	0.509301265734333
D_{3h}	0.636269323766171	D_{3m}	0.561674862453015	D_{3t}	0.527244214226719
r_{st3h}	0.604152503515955	r_{st3m}	0.758468045193244	r_{st3t}	0.837422718264700
		A_{31}	1.000000000000000		
		r_{ht3}	0.684734916442117		
		σ_{3m}	1.250000000000000		
		$f\sigma_3$	0.800000000000000		
		R_{32}	1.025900000000000		
		ϕ_{32}	1.037100000000000		

Table C15. Stator Results

ROTOR		STATOR			
ω_{SFTC2}	0.015092502816179	ω_{SFTC3}	0.052221964041667	Π_C	1.674703502643810
ω_{S2}	0.000415286496133	ω_{S3}	0.000000000000000	τ_C	1.170653666750960
ω_{P2}	0.025621919830276	ω_{P3}	0.025342625084949	η_C	1.170653666750960
ω_{T2}	0.041129709142588	ω_{T3}	0.077564589126615		
T_1/T_{t1}	0.957618541339988	P_{t3}/P_{t1}	1.674703502643810		
P_1/P_{t1}	0.859357579736388	T_{t3}/T_{t1}	1.170653666750960		
T_{R1}/T_{t1}	1.15465694167410	T_3/T_{t1}	1.128992353758110		
P_{R1}/P_{t1}	1.654190026081820	P_3/P_{t1}	1.475220458598720		
T_{E1}/T_{t1}	1.176636738281170	Φ_{3m}	0.172309288336835		
P_{E1}/P_{t1}	1.767048277556160	A_3/A_1	0.692686802248097		
P_{E2}/P_{t1}	1.734357050220880	A_3 (in)	48.440887367635200		
P_{t2}/P_{t1}	1.703686175109830				
T_2/T_{t1}	1.090701412223770				
P_2/P_{t1}	1.330027617869200				
Φ_{2m}	0.218975111752094				
A_2/A_1	0.769783662636916				
A_2 (in)	53.832415426741300				

Table C16. Stage Performance Results

ROTOR		STATOR	
H_2	1.936210985105210	H_3	1.698305724049770
C_2'	1.613509154254340	C_3'	1.415254770041480
Z_2'	26.191679106060200	Z_3'	25.192519448425300
Z_2	22.000000000000000	Z_3	27.000000000000000
AR'	1.007953705186150	AR'	1.277177533055250
C_2	1.920932454678200	C_3	1.320512345138770
S_2	1.263771351761970	S_3	1.056409876111010
i^*_{2h}	3.027273612963370	i^*_{3h}	-14.428782843997100
i^*_{2m}	6.757152202149710	i^*_{3m}	-1.724296488710650
i^*_{2t}	10.291562745234000	i^*_{3t}	3.241975399750170
ϕ^*_{2h}	-42.100896728952500	ϕ^*_{3h}	76.083582934974900
ϕ^*_{2m}	-24.523123063313700	ϕ^*_{3m}	26.611448616369400
ϕ^*_{2t}	-19.983188007865800	ϕ^*_{3t}	8.263624315276530
δ^*_{2h}	-5.757137628140870	δ^*_{3h}	21.988386237067600
δ^*_{2m}	-2.833187565346610	δ^*_{3m}	8.688685825842080
δ^*_{2t}	-1.152333894067850	δ^*_{3t}	4.391711979088870

Table C17. Blade Geometry Results

APPENDIX D. COMPARISON OF RESULTS

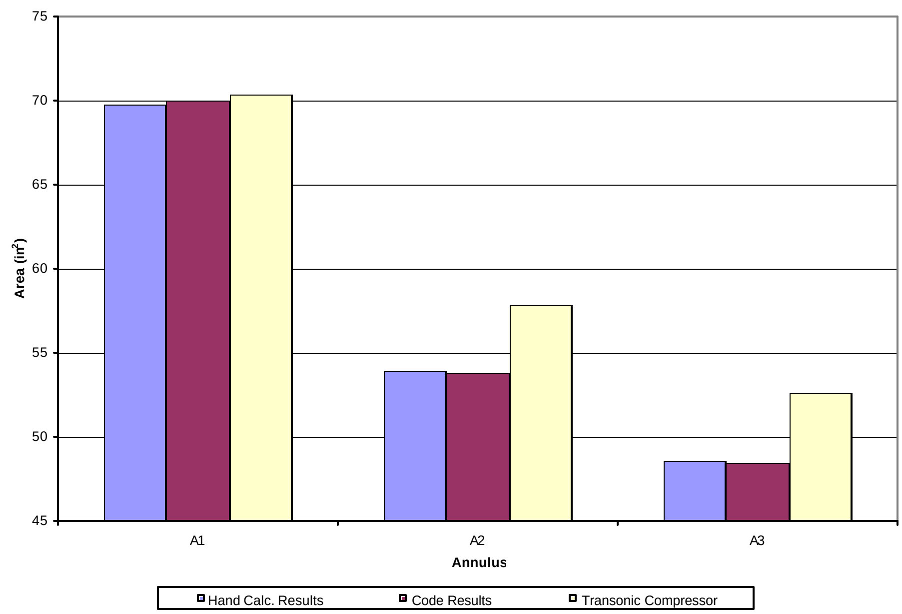

Figure D1. Annulus Comparison

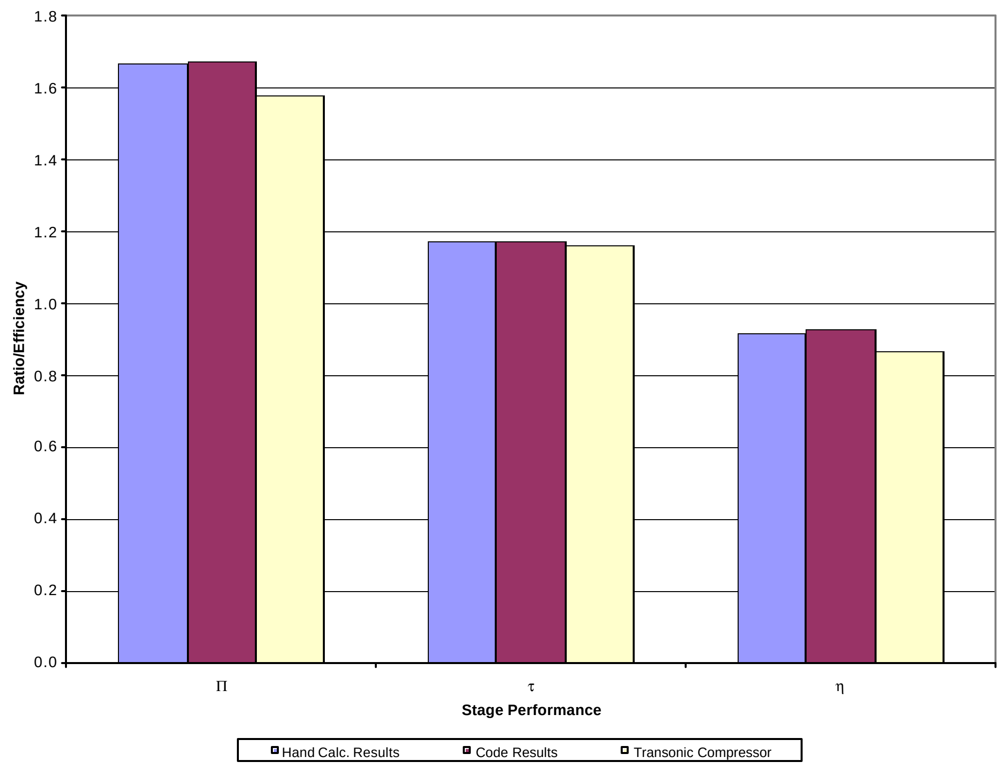

Figure D2. Stage Performance Comparison

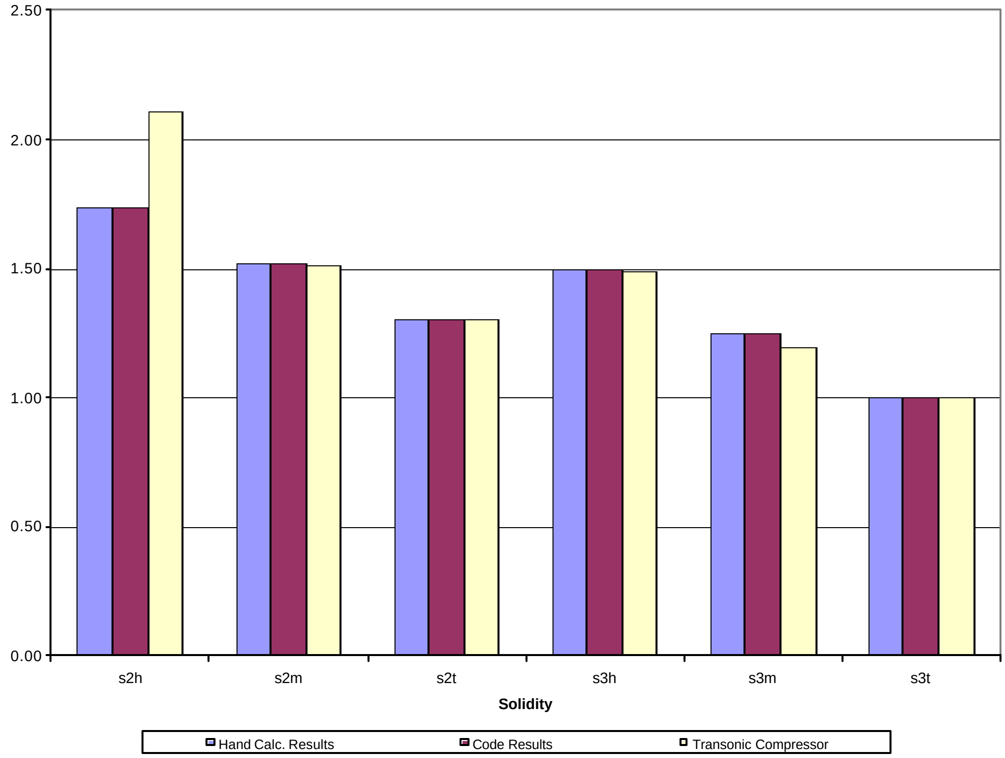

Figure D3. Solidity Comparison

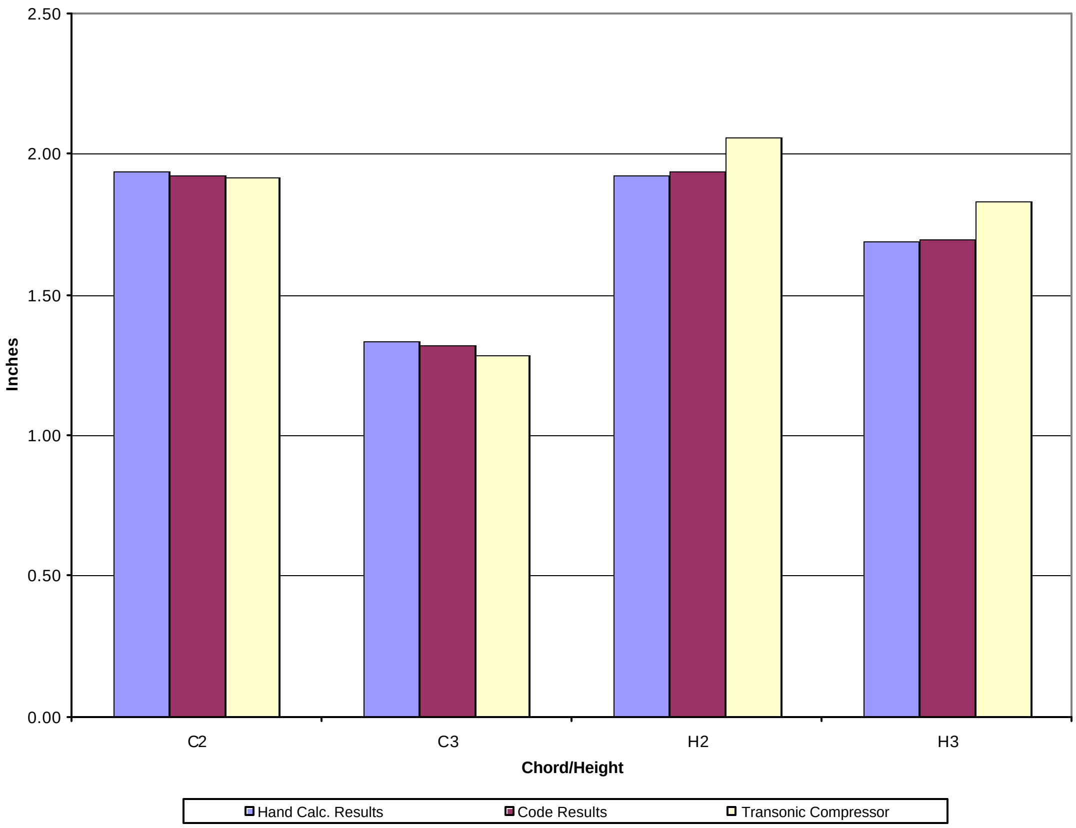

Figure D4. Chord and Blade Height Comparison

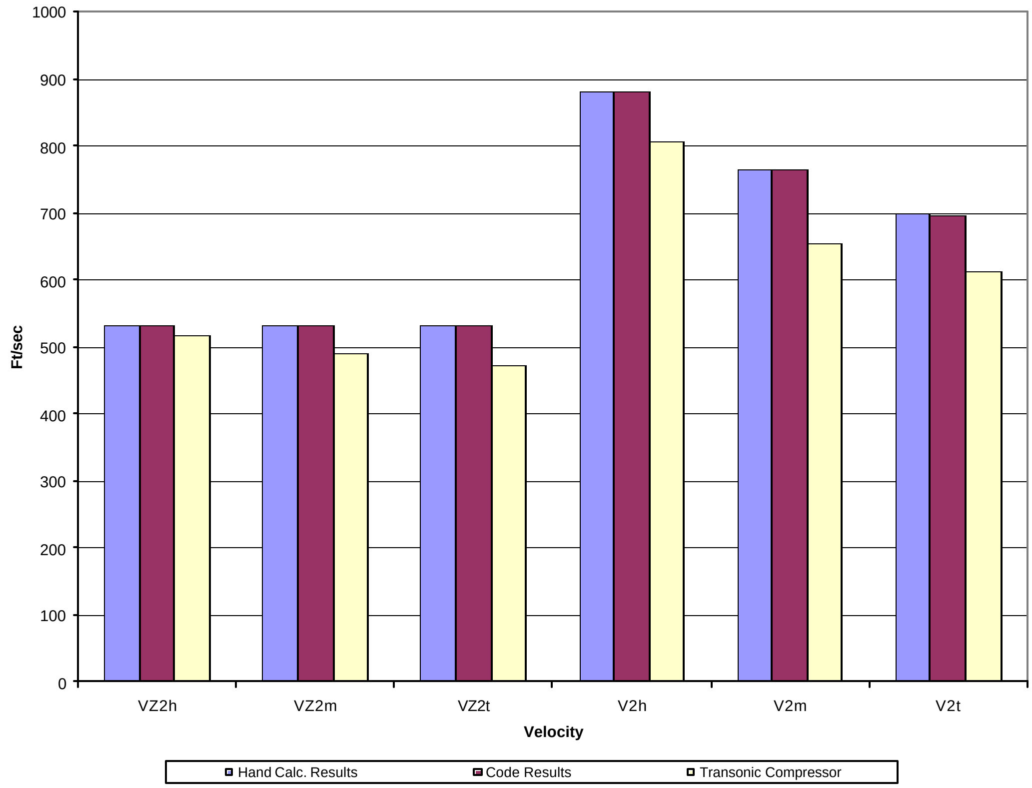

Figure D5. Rotor Velocity Comparison

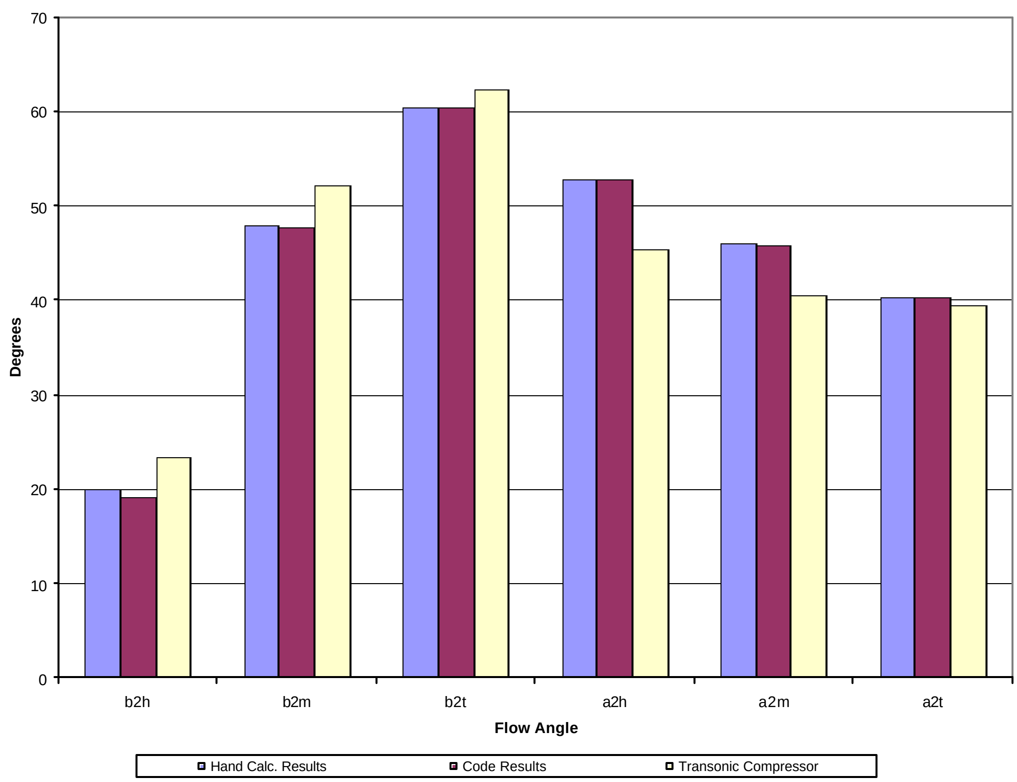

Figure D6. Rotor Flow Angle Comparison

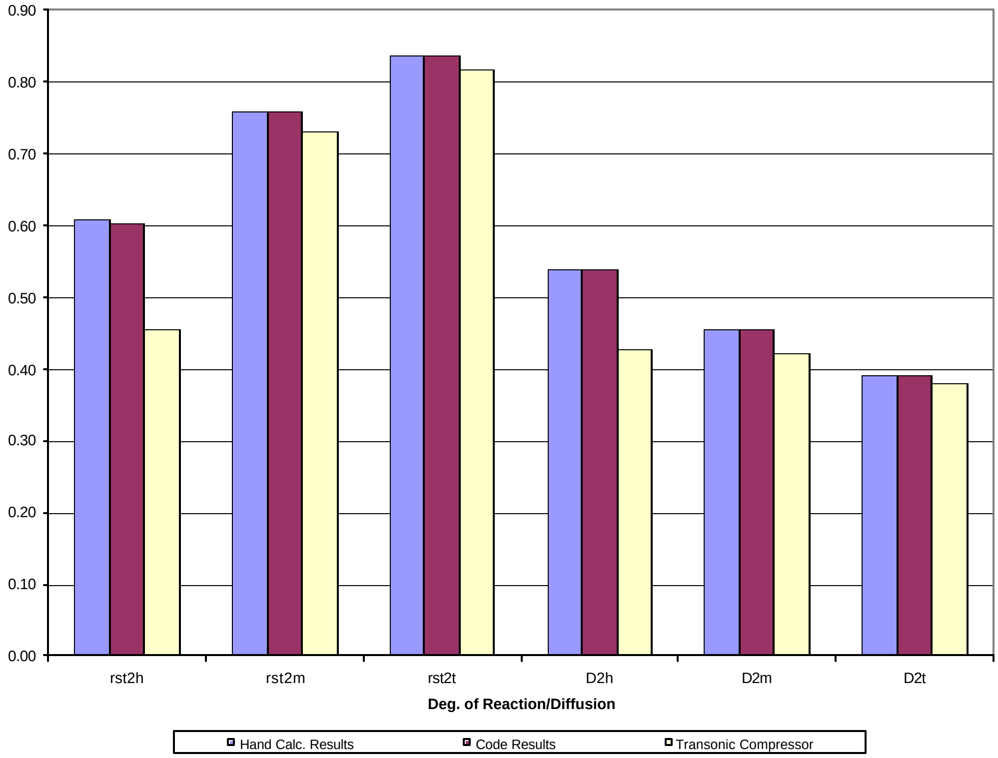

Figure D7. Rotor Degree of Reaction and Diffusion Comparison

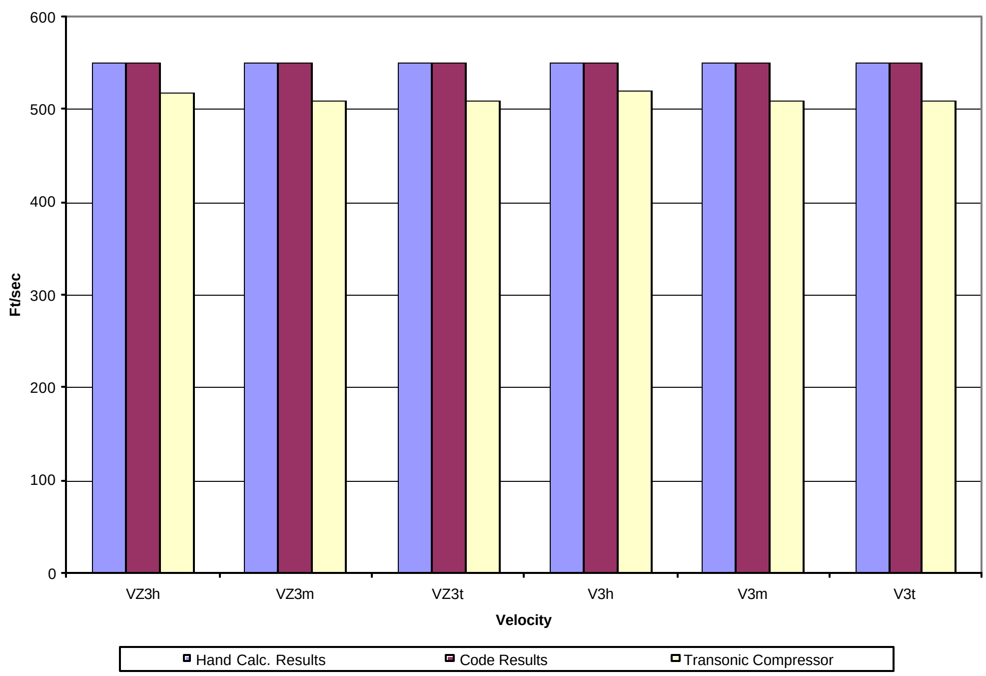

Figure D8. Stator Velocity Comparison

LIST OF REFERENCES

1. National Aeronautics and Space Administration Technical Paper 1946, *Computer Program for Aerodynamic and Blading Design of Multistage Axial-Flow Compressors*, by J.E. Crouse and W.T. Gorrell, Lewis Research Center, Cleveland, OH, 1981

2. National Aeronautics and Space Administration Technical Note D-5473, *A Computer Program for Composing Compressor Blading from Simulated Circular-Arc Elements on Conical Surfaces*, by J.E. Crouse, D.C. Janetzke and R.E. Schwirian, Lewis Research Center, Cleveland, OH, 1969

3. Shreeve, R.P., "Impulse Stage Compressor Design Program", Source Code, Naval Postgraduate School, 1996

4. Hewlett-Packard Co., "Instrument Basic for Windows", Version A.00.00, 1992

5. Microsoft Corp., "Microsoft Visual Basic 6.0 (SP5)", Version 8988, 2000

6. Kurzke, J., "GasTurb 9.0", Version 9.0, 2001

7. Shreeve, R.P., "Preliminary Design Equations", Unpublished Notes, Naval Postgraduate School, Monterey, CA, 2000

8. Shreeve, R.P., "Turbomachines – Analysis, Design & Experiment: Notes & Reference Materials for AA4431", Class Notes, Naval Postgraduate School, Monterey, CA, 1995

9. Koch, C.C. and Smith, Jr., L.H., "Loss Sources and Magnitudes in Axial-Flow Compressors", *Transactions of the ASME, Journal of Engineering for Power*", pp. 411-424, 1976

10. National Aeronautics and Space Administration SP-36, *Aerodynamic Design of Axial-Flow Compressors*, by I.A. Johnsen and R.O. Bullock, Lewis Research Center, Cleveland, OH, pp. 183-226, 1965

11. Naval Postgraduate School NPS-57SF73071A, *Calibration of Flow Nozzles Using Traversing Pitot-Static Probes*", by R. P. Shreeve, pp. 21-28, 1973

12. Advisory Group for Aerospace Research and Development AGARD-R-745, *Application of Modified Loss and Deviation Correlations to Transonic Axial Compressors*, by M. Çetin, A.S. Üçer, Ch. Hirsch, and G.K. Serovy, 1987

13. Naval Postgraduate School TPL Technical Note 99-01, *Design Methodology for the NPS Transonic Compressor*, by N.L. Sanger, Naval Postgraduate School, Monterey, CA, 1999

THIS PAGE INTENTIONALLY LEFT BLANK

BIBLIOGRAPHY

1. Dictor, E.S., *Visual Basic Controls In a Nutshell: The Controls of the Professional and Enterprise Editions*, O'Reilly and Associates, Inc., 1999

2. Jung, D., and others, *The Waite Group's Visual Basic 6 SuperBible*, Sams Publishing, 1999

3. Maiorana, Charlie, "Visual Basic Programming", Class notes, INFO/tek, Washington, D.C., 2000

4. Maiorana, Charlie, "Visual Basic II Programming", Class notes, INFO/tek, Washington, D.C., 2001

5. Oates, G.C., *Aerothermodynamics of Gas Turbine and Rocket Propulsion,* American Institute of Aeronautics and Astronautics, Inc., 1988

6. Smiley, J., *Learn to Program with Visual Basic 6.0,* Active Path Ltd., 1998

7. Wright, P., *Beginning Visual Basic 6,* Wrox Press Ltd., 1998

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   8725 John J. Kingman Road, Suite 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library
   Naval Postgraduate School

3. Chairman, Code AA
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   platzer@nps.navy.mil

4. Dr. Raymond P. Shreeve, Code AA/SF
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   shreeve@nps.navy.mil

5. Dr. Garth V. Hobson, Code AA/HG
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   gvhobson@nps.navy.mil

6. Naval Air Warfare Center – Aircraft Division
   AIR-4.4.3.3 (Attn: Mr. J. Zidzik)
   Propulsion and Power Engineering, Bldg. 106
   Zidzikjj@navair.navy.mil

7. Naval Air Warfare Center – Aircraft Division
   AIR-4.4.3.3 (Attn: Mr. M. Klein)
   Propulsion and Power Engineering, Bldg. 106
   Kleinma@navair.navy.mil

8. Naval Air Warfare Center – Aircraft Division
   AIR-4.4T (Attn: Mr. C. Gorton)
   Propulsion and Power Engineering, Building 106
   Gortonca@navair.navy.mil

9. Naval Air Warfare Center – Aircraft Division
   AIR-4.4.3.2 (Attn: Mr. R. Ramakdawala)
   Propulsion and Power Engineering, Building 3197
   Ramakdawalrr@navair.navy.mil